

**УНИВЕРЗИТЕТ „СВ.КИРИЛ И МЕТОДИЈ“
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И
КОМПЈУТЕСКО ИНЖЕНЕРСТВО**



М-р НАТАША В. СТОЈКОВИЌ

**НАДЕЖНОСТ НА
ПОВЕЌЕ – СОСТОЈБЕНИ ТРАНСПОРТНИ СИСТЕМИ И
ОПТИМАЛНО ПОДОБРУВАЊЕ НА НИВНИТЕ ПЕРФОРМАНСИ**

- ДОКТОРСКА ДИСЕРТАЦИЈА -

**СКОПЈЕ, 2015
РЕПУБЛИКА МАКЕДОНИЈА**

Ментор: Проф. д-р Марија Михова
Факултет за информатички науки и компјутерско инженерство - Скопје

Комисија: Проф. д-р Марија Михова
Факултет за информатички науки и компјутерско инженерство – Скопје

Проф. д-р Жанета Попеска
Факултет за информатички науки и компјутерско инженерство – Скопје

Проф.д-р Невена Ацковска
Факултет за информатички науки и компјутерско инженерство – Скопје

Доц. д-р Миле Јованов
Факултет за информатички науки и компјутерско инженерство – Скопје

Доц. д-р Кире Триводалиев
Факултет за информатички науки и компјутерско инженерство - Скопје

Датум на одбрана: _____

Датум на промоција: _____

Научна област: Информатика

Резиме

Цел на проучување на докторската дисертација се двотерминалните повеќе – состојбени транспортни системи, поточно наоѓање на методи и техники за пресметување на нивната надежност. Во докторската дисертација се предложени нови алгоритми за наоѓање на минимални пат вектори за двотерминални повеќе–состојбени транспортни системи. При разгледување на познатите алгоритми во литературата и нивно анализирање дојдено е до заклучок дека графот што е индуциран од минимален пат вектор е ацикличен, од каде произлезе и главната идеја за развој на новите алгоритми.

Во воведниот дел се дадени основните концепти на надежноста, дефиницијата на надежност и дефиницијата на други поими кои се користат при анализа на надежност. Исто така се дадени и основните концепти на надежност на бинарен систем и на систем со повеќе состојби. Дефиницијата на структурната функција за бинарни и повеќе – состојбени системи, како и дефинициите за минимални пат вектори и минимални пресек вектори. Во втората глава, даден е преглед на познатите техники за пресметување на надежност на двотерминалните бинарни транспортни системи. Во продолжение се предложени два пристапи за наоѓање на минимални пресек множества на двотерминален бинарен ненасочен транспортен систем. Во третата глава е дефинирана повеќе – состојбената двотерминална надежност и се разгледуваат проточните транспортни системи. На крајот на оваа глава е даден краток преглед на трудовите кои се наменети за пресметување на надежност на двотерминални повеќе – состојбени транспортни системи. Во четвртата глава е предложена теорија што ја поврзува теоријата на повеќе – состојбена надежност со теоријата на проток во двотерминални транспортни системи. Во оваа глава се анализирани својствата на минималните пат вектори, и е покажана врска помеѓу минималните пат вектори за ниво d и протокот за ниво d за двотерминален проточен транспортен систем. Во петтата глава се предложени алгоритми за наоѓање на минималните пат вектори за ниво d и за насочени и за ненасочени транспортни системи. Сите алгоритми што се предложени се базираат на фактот дека графот што е индуциран од минималниот пат вектор е ацикличен.

КЛУЧНИ ЗБОРОВИ: надежност, двотерминален транспортен систем, минимален пат вектор, минимален пресек вектор, минимално пат множество, минимално пресек множество, насочени транспортни системи, ненасочени транспортни системи, граф.

ABSTRACT

The aim of the study of this dissertation are two terminal multi – state networks, namely finding on methods and techniques for calculating of their reliability. In the doctoral dissertation new algorithms for calculating minimal path vectors in two terminal multi – state network are proposed. With the analysis of the known algorithms in the literature, it is concluded and proved that the graph which is induced of the minimal path vector is acyclic, from where the main idea for the development of the new algorithms is derived.

The introduction provides the basic concepts of reliability, more precisely the definition of reliability and the definitions of other terms used in the analysis of reliability. Also, basic concepts of reliability of a binary system and a multi – state system are given. Definition of the structure function and definitions for minimal path and minimal cut vectors are given. In the second chapter, the overview of known methods for calculating on reliability on two terminal binary network is given. Further on, two approaches to determine minimal cut set for two terminal binary undirected network are proposed. In the third chapter two terminal multi – state reliability is defined and techniques for its calculation are given. Afterwards, flow networks are considered. At the end of this chapter, there is a brief overview of the papers for calculating two terminal multi – state reliability provided. In the fourth chapter extended theory that connects the multi – state reliability theory with the theory for flow in two terminal networks is proposed. In this chapter properties of the minimal path vector are analysed, and the connection between minimal path vectors and flow functions for two terminal flow network is considered. In the fifth chapter, algorithms for calculating the minimal path vectors to level d both for directed and for undirected networks are developed. All proposed algorithms are based on the fact that the graph which is induced by minimal path vector is acyclic.

KEY WORDS: *reliability, two terminal network, minimal path vector, minimal cut vector, minimal cut set, directed network, undirected network, graph.*

Благодарност

Сакам да ја искажам мојата голема благодарност до мојот ментор проф.д-р Марија Михова за сето знаење, помош и поддршка кои ги добив од неа во текот на повеќегодишната наша соработка и се надевам дека таа ќе продолжи и понатаму.

Би сакала да се заблагодарам на проф.д-р Жанета Попеска, проф.д-р Невена Ацковска, доц. д-р Миле Јованов и доц.д-р Кире Триводалиев за корисните сугестии кои ги добив од нивна страна.

Посебна благодарност упатувам кон мојот сопруг кој постојано ме поддржуваше, а уште повеќе кон мојата ќерка која имаше големо разбирање за мојата ангажираност во текот на изработката на оваа докторска дисертација.

Благодарност до моето семејство, моите пријатели и колегите од Факултетот за информатика при Универзитетот “Гоце Делчев” – Штип за моралната поддршка при изработката на дисертацијата.

СОДРЖИНА

1. ВОВЕД	1
1.1 НАДЕЖНОСТ НА БИНАРНИ СИСТЕМИ	4
1.2 НАДЕЖНОСТ НА СИСТЕМИ СО ПОВЕЌЕ СОСТОЈБИ	5
2. НАДЕЖНОСТ НА БИНАРНИ ТРАНСПОРТНИ СИСТЕМИ	9
2.1 ГРАФИЧКО ПРЕТСТАВУВАЊЕ НА ТРАНСПОРТНИОТ СИСТЕМ	11
2.1.1 ГРАФ	11
2.1.2 ПРЕТСТАВУВАЊЕ СО ГРАФ	11
2.2 НАДЕЖНОСТ НА БИНАРНИ ТРАНСПОРТНИ СИСТЕМИ	12
2.2.1 АНАЛИЗА НА НАДЕЖНОСТ НА ТРАНСПОРТНИ СИСТЕМИ	13
2.2.2 ПРЕГЛЕД НА МЕТОДИТЕ ЗА ПРЕСМЕТУВАЊЕ НА НАДЕЖНОСТ НА БИНАРНИ ТРАНСПОРТНИ СИСТЕМИ	14
2.2.2.1 Метод со набројување на сите можни состојби	16
2.2.2.2 Метод на декомпозиција (факторизација) (Factoring method)	16
2.2.2.3 Метод со примена на формула за вклучување и исклучување	18
2.2.2.4 Метод со сума од дисјунктни производи (SDP)	19
2.3 ОПРЕДЕЛУВАЊЕ НА МИНИМАЛНИ ПРЕСЕК МНОЖЕСТВА ВО БИНАРЕН ТРАНСПОРТЕН СИСТЕМ	20
2.3.1 АЛГОРИТАМИ ЗА ОПРЕДЕЛУВАЊЕ НА МИНИМАЛНИ ПРЕСЕК МНОЖЕСТВА НА ТРАНСПОРТНИ СИСТЕМИ	21
2.3.2 АЛГОРИТАМ ЗА ОПРЕДЕЛУВАЊЕ НА МИНИМАЛНИ ПРЕСЕК МНОЖЕСТВА НА НЕНАСОЧЕНИ МРЕЖИ СО КОРИСТЕЊЕ НА BFS	31
3. НАДЕЖНОСТ НА ПОВЕЌЕ - СОСТОЈБЕНИ ТРАНСПОРТНИ СИСТЕМИ И ПРОТОЧНИ ТРАНСПОРТНИ СИСТЕМИ	39
3.1 НАДЕЖНОСТ НА ПОВЕЌЕ - СОСТОЈБЕНИ ДВОТЕРМИНАЛНИ ТРАНСПОРТНИ СИСТЕМИ	39
3.1.1 ПРЕСМЕТУВАЊЕ НА НАДЕЖНОСТ НА ПОВЕЌЕ – СОСТОЈБЕНИ ТРАНСПОРТНИ СИСТЕМИ	41
3.2 ПРОТОЧНИ ТРАНСПОРТНИ СИСТЕМИ	42
3.2.1 АЛГОРИТАМ НА FORD - FULKERSON	45
3.2.2 ОСТАТОЧЕН ТРАНСПОРТЕН СИСТЕМ (RESIDUAL NETWORK)	46
3.2.3 ЗГОЛЕМУВАЧКИ ПАТ	47
3.2.4 ПРЕСЕЦИ ВО ПРОТОЧНИ МРЕЖИ	47
3.3. ПРЕГЛЕД НА ИСТРАЖУВАЊАТА НА НАДЕЖНОСТА НА ДВОТЕРМИНАЛНИ ПОВЕЌЕ-СОСТОЈБЕНИ ТРАНСПОРТНИ СИСТЕМИ	49
4. ВРСКА ПОМЕЃУ МИНИМАЛЕН ПАТ ВЕКТОР И ПРОТОК КАЈ ДВОТЕРМИНАЛНИ ПРОТОЧНИ ТРАНСПОРТНИ СИСТЕМИ	55

4.1 Врска помеѓу минималните пат вектори и функциите на проток кај двотерминални проточни насочени транспортни системи	56
4.2 Врска помеѓу минималните пат вектори и функциите на проток кај двотерминални проточни ненасочени транспортни системи	60
5. АЛГОРИТАМИ ЗА ПРЕСМЕТУВАЊЕ НА МИНИМАЛНИ ПАТ ВЕКТОРИ ЗА НИВО D БАЗИРАНИ НА СВОЈСТВОТО НА АЦИКЛИЧНОСТ НА ГРАФОВИ ИНДУЦИРАНИ ОД ПРОТОК	71
5.1 Алгоритам со контрола на циклуси	79
5.2 Алгоритаи за наоѓање на минимални пат вектори за насочени транспортни системи со користење на циклуси во остаточниот транспортен систем	91
5.3 Подобрен алгоритам за наоѓање на минимални пат вектори за ненасочени транспортни системи	103
6.ЗАКЛУЧОК	109
КОРИСТЕНА ЛИТЕРАТУРА	111
ПРИЛОГ А	115
Алгоритаи за наоѓање на минимални пат вектори предложени од Y.S. LIN, C.C JANE и J. YUAN и M. FORGHANI – ELAHABAD и N. MAHDAVI-AMIRI	115
ПРИЛОГ Б	129
Алгоритам за наоѓање на минимални пат вектори предложен од M. MIHOVA и N. SYNAGINA	129

Глава 1

1.Вовед

Надежноста претставува една од најзначајните карактеристики за квалитетот на еден систем, бидејќи системот мора да работи задоволително пред да се оценат другите квалитативни карактеристики. Надежноста на системот може да се сфати како веројатност дека системот ќе функционира во текот на даден временски период под одредени услови. Во математичката теорија на надежноста, оценувањето на надежноста може да се дефинира како проблем кој се состои од две фази. Во првата фаза се анализира како однесувањето на компонентите на системот влијае врз работата на системот. Оваа фаза е клучна фаза бидејќи во неа се разгледува како компонентите од кои е составен системот влијаат на неговото однесување. Во втората фаза се пресметува надежноста на системот. Базирајќи се на заемното дејство на компонентите во системот, како и врз основа на нивните квантитативни карактеристики, за надежноста на системот може да се добијат точни или апроксимативни вредности [6].

Кај комплексните системи е многу е тешко да се постигне висока надежност. Од овие причини пред да се започне со дизајн на еден комплексен систем потребно е да се формулираат различни пристапи со кои ќе се истакнат предностите и недостатоците што може да се појават доколку се користи тој дизајн.

Предмет од интерес на овој труд се транспортните системи. Нас пред сè не интересира како надежноста на еден систем зависи од неговите составни компоненти, на што ќе дадеме посебен осврт. Но, сепак, во воведниот дел ќе ги дадеме и основните дефиниции за надежност на еден систем, во случај кога неговата работа се разгледува како процес кој се менува во текот на времето. Во овој случај работа на системот како и можноста за негово расипување зависи од времето на работа. Со помош на математички апарат надежноста може да се запише како функција која зависи од времето на работа на системот. Јасно е дека во текот на работа системот може да се расипе, поточно одреден временски период тој може да работи задоволително, а после тој период да не функционира со задоволително ниво. Со случајната променлива T да го означиме времето на живот на системот. Во некој момент t системот може да расипе, всушност t претставува

вредност на случајната променлива T . Функцијата на распределба на случајната променлива T е:

$$F(t) = P(T \leq t). \quad (1.1)$$

$F(t)$, се нарекува ненадежност во момент t . Тоа ја претставува веројатноста да се случи грешка во интервалот $[0, t]$. Веројатноста за да се појави грешка во интервалот $(t_1, t_2]$ е еднаква на $F(t_2) - F(t_1)$.

Дефиниција 1.1 Надежност на системот се дефинира со:

$$R(t) = P(T > t) = 1 - F(t). \quad (1.2)$$

т.е. надежноста на системот е веројатноста дека нема да се случи грешка, т.е. дека системот нема да се расипе во интервалот $[0, t]$, или веројатност дека системот ќе се расипе после време t .

Случајната променлива T може да биде дискретна случајна променлива или непрекината случајна променлива. Во случај кога T е дискретна случајна променлива таа може да прими конечно или преброиво многу вредности. Времето на расипување на системот може да се случи во некоја од вредностите што може да ги прими T . Меѓутоа во најголем број случаи, T е непрекината случајна променлива со непрекината функција на распределба $F(t)$ и густина на распределба $f(t)$.

Накратко да ги разгледаме двата случаја кога T е дискретна и кога T е непрекината случајна променлива.

Да претпоставиме дека T е дискретна случајна променлива. Тогаш T напoлно ќе биде определна со множеството од вредности $0 = t_0 < t_1 < t_2 < \dots$, што може да ги прима и со законот на распределба на веројатностите во тие вредности:

$$p(t_i) = P(T = t_i), \quad i = 0, 1, 2, \dots \quad (1.3)$$

Вообичаено се заменува $t_i = i$, и за надежноста се добива:

$$R(t_i) = P(T > t_i) = p(i + 1) + p(i + 2) + \dots, \quad (1.4)$$

во општ случај

$$R(t) = P(T > t) = \sum_{i: t_i > t} p(t_i). \quad (1.5)$$

Да забележиме дека:

$$p(t_i) = R(t_{i-1}) - R(t_i). \quad (1.6)$$

Ако T е непрекината случајна променлива, која прима вредности од интервалот $(0, \infty)$, и има густина на распределба $f(t)$, тогаш за надежност се добива :

$$R(t) = \int_t^{\infty} f(t)dt = 1 - \int_0^t f(t)dt = 1 - F(t), \quad (1.7)$$

каде за густината на распределба важи:

$$f(t) \geq 0, \quad \int_0^{\infty} f(t)dt = 1,$$

$$f(t) = -\frac{dR(t)}{dt}$$

Значи веројатноста дека системот ќе се расипе во временскиот интервалот (t_1, t_2) , е:

$$P(t_1 < T < t_2) = F(t_2) - F(t_1) = R(t_1) - R(t_2). \quad (1.8)$$

При анализа на надежност на системи, од интерес е да се разгледува и очекуваното време да се расипе одредена компонента на системот или целиот систем, ова време се нарекува **средното време на расипување (MTTF)**.

$$\text{MTTF} = E(T) = \int_0^{\infty} tf(t)dt = \int_0^{\infty} R(t)dt. \quad (1.9)$$

Средното време помеѓу две расипувања **MTBF** е очекуваното време да се расипе системот, откако некоја компонента или системот веќе биле расипани и поправени. Оваа мерка се разгледува кај системи кои може да се поправаат.

Други значајни мерки на системите се и одржливост $V(t)$, достапност $\mathcal{A}(t)$ и средно време на поправка.

Одржливост се дефинира како веројатност дека системот што не работи ќе биде вратен во функционална состојба во даден временски период, кога одржувањето е извршено во согласност со одредени процедури и ресурси. Ако со T_1 го дефинираме **времето на поправка** кое има густина на распределба $g(t)$, тогаш одржливост е всушност веројатноста системот што не работи да се врати во работна состојба за време t :

$$V(t) = P(T_1 < t) = \int_0^t g(x)dx. \quad (1.10)$$

Очекуваната вредност од времето на поправка се нарекува средно време на поправка (MTTR).

За средното време на расипување, средното време помеѓу две расипувања и средното време на поправка важи следното: $\text{MTBF} = \text{MTTF} + \text{MTTR}$.

Функцијата на достапност, означена со $\mathcal{A}(t)$ се дефинира како веројатност дека системот е во работна состојба во време t .

$$\mathcal{A}(t) = P(\text{Системот е во работна состојба во временски момент } t).$$

Во наредните поглавја ќе ја дефинираме надежноста на бинарните системи и надежноста на повеќе - состојбните системи.

1.1 Надежност на бинарни системи

Ако компонентите во системот и самиот систем можат да се најдат само во две состојби: компонентата (или системот) работи и компонентата (или системот) е расипана, тогаш за таквиот систем велиме дека е бинарен систем. Во првиот дел од ова поглавје ќе ги дадеме воведните поими за бинарни системи, [4].

Вообичаено, во математичката теоријата на надежност се претпоставува дека системот и неговите компоненти може да се најдат во една од две можни состојби т.е. можат да бидат во работна состојба или да бидат расипани. Постојат голем број на алгоритми за пресметување на надежноста на различни бинарни системи. Сепак не постои универзален пристап кој може да се примени за сите бинарни системи.

При анализата на надежност на системите, се оценува како состојбата во која се наоѓаат компонентите на системот влијае врз состојбата во која се наоѓа системот и обратно како состојбата во која се наоѓа системот влијае врз состојбата во која се наоѓаат неговите компоненти. Поради тоа што системот во целина и неговите компоненти имаат бинарна природа, тогаш секоја компонента може да биде претставена со бинарна случајна променлива x_i , каде $x_i = 1$ ако компонентата i е во работна состојба и $x_i = 0$ ако компонентата е расипана. Состојбата на системот ја означуваме со φ , при што $\varphi = 1$ ако системот работи и $\varphi = 0$ ако системот е расипан. Ако претпоставиме дека системот има n компоненти, тогаш векторот $\vec{x} = (x_1, x_2, \dots, x_n)$ ја опишува состојбата на сите компоненти на системот и се нарекува **вектор од состојби на системот**. Состојбата во која се наоѓа системот во целина зависи од составните делови на системот, па се дефинира функција во која состојбата на системот се изразува преку состојбата на неговите компоненти. Ова функција се нарекува структурна функција и се дефинира на следниот начин:

$$\varphi(x_1, x_2, \dots, x_n) = \varphi. \quad (1.11)$$

Дефиниција 1.2 Пат е множество од компоненти за кои важи дека, ако тие компоненти работат тогаш системот работи.

Дефиниција 1.3 Пресек е множество од компоненти за кое важи дека ако компонентите од тоа множество се расипани, тогаш и целиот систем е расипан.

Дефиниција 1.4 Еден пат е минимален пат ако не постои негово вистинско подмножество кое исто така е пат.

Дефиниција 1.5 Еден пресек е **минимален пресек** ако не постои негово вистинско подмножество кое исто така е пресек.

Во продолжение, е даден еден начин на пресметување на надежноста. Ако минималните патишта или минималните пресеци се познати тогаш со помош на добро познатата формула за вклучување и исклучување може да се пресмета надежноста на системот.

Нека P_1, P_2, \dots, P_l се минимални патишта, тогаш надежноста се пресметува со следнава формула:

$$R = P\left(\bigcup_{h=1}^l P_h\right) = \sum_{h=1}^l P(P_h) - \sum_{h < k} P(P_h \cap P_k) + \dots + (-1)^l P(P_1 \cap \dots \cap P_l). \quad (1.12)$$

каде l е бројот од минимални патишта и P_h е минималниот пат h .

Во случај кога се познати минималните пресеци C_1, C_2, \dots, C_w , тогаш надежноста може да се пресмета со следната формула:

$$R = 1 - P\left(\bigcup_{h=1}^w C_h\right) = 1 - \left(\sum_{h=1}^w P(C_h) - \sum_{h < k} P(C_h \cap C_k) + \dots + (-1)^w P(C_1 \cap \dots \cap C_w)\right). \quad (1.13)$$

каде w е бројот од минимални патишта и C_h е минималниот пресек h .

1.2 Надежност на системи со повеќе состојби

За разлика од бинарните системи постојат системи кои имаат компоненти што работат во повеќе нивоа на работа, поточно квалитетот на работа на компонентата може да опадне, но компонентата да не се расипе целосно. Ние разгледуваме компоненти кои имаат конечен број нивоа на работа. При тоа нивоата на работа како вредности може да примат кој било реален број. Значи доколку случајната променлива x_i има k нивоа во кои работи, и едно во кое не работи таа ќе може да прима вредности од множеството

$$S_i = \{0, m_{i_1}, m_{i_2}, \dots, m_{i_k}\}, \text{ каде } 0 < m_{i_1} < m_{i_2} < \dots < m_{i_k} \text{ и } m_{i_j} \in \mathbb{R} \text{ за } j = 1, 2, \dots, k.$$

Состојбата m_{i_k} ја нарекуваме **перфектно ниво** и обично перфектното ниво за i -тата компонента се означува со M_i . Со 0 го означуваме нивото во кое компонентата е тотално расипана. Па може да запишеме дека $0 \leq x_i \leq M_i$, $x_i \in \mathbb{R}$. Множеството S_i ги содржи сите можни состојби во кои може да се најде i -тата компонента [4].

Ако системот има n -компоненти тогаш случајниот вектор $\vec{x} = (x_1, x_2, \dots, x_n)$, $x_i \in S_i$, $i=1, \dots, n$, што ги опишува состојбите на сите компоненти на системот се нарекува **вектор од состојби на системот**. Ова е всушност состојбата на секоја од компонентите од кои е составен системот. Множеството од сите состојби ќе го бележиме со \mathcal{E} , и $\mathcal{E} = S_1 \times S_2 \times \dots \times S_n$. Целиот систем, исто како и неговите компоненти може да работи со различен квалитет, така што во повеќе состојби да има ист квалитет. Квалитетот на работа на

системот се нарекува **ниво на работа**. Множеството од нивоа на работа на системот се бележи со $S = \{m^1, m^2, \dots, m^k\}$ каде m^i , за $i=1, \dots, k$ се можните нивоа на работа во кои може да се најде целиот систем. Максималното ниво на работа на системот обично го означуваме со M .

Нивото на работа на системот во зависност од неговите компоненти, исто како и кај бинарните системи, може да се претстави преку структурна функција.

Дефиниција 1.6 Нека S_i е множество на состојби на i -тата компонента во n компонентен систем, а S множество нивоа на работа на системот. Функцијата $\varphi : S_1 \times S_2 \times \dots \times S_n \rightarrow S$, дефинирана со

$$\varphi = \varphi(\vec{x}). \quad (1.14)$$

се нарекува **структурна функција**.

Кај бинарните системи состојбите во кои системот работи се претставуваат со пат множества, а состојбите во кои тој не работи со пресек множества. Додека пак минималните перформанси што треба да ги задоволи системот за да работи се претставуваат со минимални пат множества, а максималните перформанси при кои системот не работи со пресек множества. Аналогно на овие множества, кај повеќе – состојбените системи се дефинираат пат, пресек, минимален пат и минимален пресек вектори.

За да ги дефинираме овие вектори, прво ќе дефинираме подредување на множеството состојби на системот.

Дефиниција 1.7 За векторот \vec{y} , велиме дека е помал од векторот \vec{x} , $\vec{y} < \vec{x}$ (или доминиран од \vec{x}) ако и само ако $\forall i, y_i \leq x_i$ и за некое $k, y_k < x_k$.

Како што кажавме претходно, нивото на работа на системот се претставува со структурна функција φ , дефинирана на множеството \mathcal{E} . Со помош на оваа функција ќе дефинираме пат и пресек вектори.

Дефиниција 1.8 Векторот $\vec{x} \in \mathcal{E}$ е **пат вектор за ниво d** , ако $\varphi(\vec{x}) \geq d$.

Дефиниција 1.9 Векторот $\vec{x} \in \mathcal{E}$ е **пресек вектор за ниво d** ако $\varphi(\vec{x}) < d$.

Векторот \vec{x} е пат вектор за ниво d , ако системот во состојба \vec{x} работи со ниво поголемо или еднакво на d . Додека векторот \vec{x} е пресек вектор за ниво d , ако системот во состојба \vec{x} работи со ниво помало од d .

Дефиниција 1.10 Векторот \vec{x} е **минимален пат вектор за ниво d** ако е пат вектор за ниво d и не постои друг пат вектор за ниво d кој е строго помал од него.

Дефиниција 1.11 Векторот \vec{x} е **минимален пресек вектор за ниво d** ако е пресек вектор за ниво d и не постои друг пресек вектор за ниво d кој е строго поголем од него.

Множеството од сите минимални пат вектори за ниво d ќе го означиме со \mathcal{P}_d , а множеството од сите минимални пресек вектори за ниво d со \mathcal{C}_d .

Веројатноста системот да работи со ниво поголемо или еднакво од d е веројатноста тој да се најде во вектор состојба поголема или еднаква на некој од векторите од множеството \mathcal{P}_d .

$$R_d = P(\varphi(\vec{x}) \geq d). \quad (1.15)$$

Глава 2

2. Надежност на бинарни транспортни системи

Проучувањето на надежноста на комплексните системи, т.е. на системите кои содржат прилично голем број на компоненти кои се поврзани помеѓу себе е од посебен интерес за луѓето од различни области: од техничките области до биолошките и економските области. Да нагласиме дека многу комплексни физички, технолошки, социјални, биолошки и економски системи може да се претстават со форма на транспортни системи, каде јазлите се објекти на системот, додека линковите ги претставуваат врските помеѓу објектите. Едно битно својство на транспортните системи, кое ги прави да бидат преферирани во природните и техничките науки, е тоа што врската меѓу два јазла од транспортниот систем може вообичаено да се постигне преку неколку редундантни патишта, а тоа ја прави врската сигурна.

Од почетокот на 60-тите години на минатиот век, поради експоненцијален раст на големината и бројот на компонентите во сите видови на транспортни системи, многу истражувачи започнуваат да ги проучуваат карактеристиките, топологијата, и надежноста на транспортните системи.

Во последните години, анализата на надежноста на транспортните системи е главна грижа при креирањето, валидација и одржување на многу реални системи, како комуникациските, компјутерските или електричните транспортни системи. Надежноста на овие комплексни системи е од големо значење со оглед на тоа дека расипувањето на некои компоненти на системот може да доведе до катастрофални резултати.

Според стандардот IEEE 90 надежноста се дефинира како „способност на системот и неговите компоненти да функционираат успешно под одредени услови за одреден период на време”.

Поради оваа причина, важно е да се обезбедат методи со кои може да се утврди со која веројатноста системот или некоја негова компонента ќе се расипе. На пример, расипувањето на електричната мрежа во Северна Америка на 14 август 2003 оставило без електрично напојување повеќе од 50 милиони луѓе во САД и Канада. Расипувањето на некои компоненти на системот за пренос на енергија во државата Охајо довело до

расипување на 22 нуклеарни и 80 термоцентрали за помалку од 10 секунди. Во телекомуникациските транспортни системи максимално прифатливо ниво транспортниот систем да не работи е приближно 5 минути во текот на една година [48].

Надежноста на системот зависи од надежноста на компонентите од кои е составен системот и од топологијата (структурата) на системот. Затоа при анализа на надежност на системите, ги земаме предвид можните расипувања на компонентите на системот (линкови или јазли), знаејќи дека овие расипувања може да доведат до расипување на системот во целина.

Транспортниот систем може да се претстави со помош на граф. Елементите на графот, и јазлите и линковите може да се наоѓаат во една од две состојби (да работат или да се расипани) – кога станува збор за бинарни транспортни системи, или пак да се наоѓаат во една од повеќе можни состојби (може да бидат расипани, да работат со полн капацитет, или делумно да работат) – кога станува збор за повеќе – состојбени транспортни системи. Во овој труд ние ќе ги анализираме повеќе – состојбените транспортни системи и ќе претпоставиме дека јазлите не се расипуваат (работат со полн капацитет), додека линковите може да се најдат во една од повеќето можни состојби. Бидејќи нè интересира надежноста на системот во зависност од неговите составни делови, можеме времето да го занемариме, или пак да го специфицираме, при што во секој момент веројатноста дека системот ќе работи во одредено ниво на работа ќе биде еднаква.

За да го запазиме континуитетот на оваа теорија, прво ќе ги разгледаме бинарните транспортни системи. Востановено е дека бинарните системи недоволно ја опишуваат работата на некои транспортни системи, па заради тоа се преминува кон теоријата на повеќе – состојбени системи.

Како што кажавме, транспортните системи можеме да ги претставиме со помош на графови, па надежноста на транспортните системи може да се дефинира како веројатност дека сите јазли на графот, или подмножеството од јазли, се поврзани со најмалку еден пат.

При пресметувањето на надежноста се користат техники со кои се претпоставува дека елементите на транспортниот системи (и јазлите и линковите) се статистички независни. Техниките за пресметување на надежноста се класифицираат во две главни категории:

1. Пристапи во кои надежноста на транспортните системи се пресметува директно,
2. Пристапи со кои прво се набројуваат сите можни поврзувања помеѓу јазлите и потоа се оценува надежноста.

Ние се одлучивме за вториот пристап за пресметување на надежноста, поточно за оценување на надежноста ќе ја користиме формулата за вклучување и исклучување (1.12) и (1.13). Овој пристап ќе биде подетално опишан и во едно од следниве подглавја.

Во многу случаи, анализата на перформансите на системот бара на линковите да им се доделат одредени тежини, кои може да претставуваат капацитет на линкот, цена на линкот или пак должина на линкот. Во нашиот труд линковите секогаш ќе бидат дадени со нивните капацитети.

2.1 Графичко претставување на транспортниот систем

Транспортниот систем е систем којшто најдобро математички може да се претстави со помош на граф. Темињата на графот ги претставуваат елементите на системот, додека ребрата (линковите) ги претставуваат врските (релациите помеѓу елементите). Во општ случај линкови би биле телефонски кабел за телекомуникациски систем, водоводни цевки за систем за пренос на вода, гасоводни цевки за систем за пренос на гас, и слично. Да нагласиме дека термините што се користат за опишување на графови не се унифицирани. Најчесто се користат термините теме и лак, јазли и гранки или места и меѓусебни врски, линкови. Во оваа теза ќе ги користиме термините јазли и линкови.

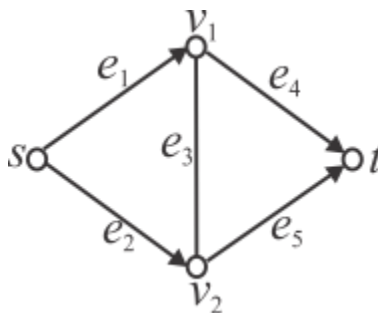
2.1.1 Граф

Транспортниот систем може да се претстави во форма на граф $G(V, E)$, каде V е множеството од јазли додека E е множеството од линкови. Нека со n го означиме бројот на линкови во системот, а со m бројот на јазли во системот. Еден линк е насочен ако комуникацијата помеѓу јазлите што ги поврзува се одвива во една насока. Ако во графот постои барем еден насочен линк тогаш станува збор за **насочен граф**. Еден линк е ненасочен ако комуникацијата помеѓу јазлите што ги поврзува може да се одвива во двете насоки. Ако во графот сите линкови се ненасочени станува збор за **ненасочен граф**. За ненасочени транспортни системи без јамки (ние ќе разгледуваме графови без јамки поради тоа што ваквите линкови не пренесуваат информации од едно место до друго), постоењето на најмалку еден линк што го поврзува секој јазол во системот, гарантира дека сите јазли се поврзани. Додека во насочените графови, јазлите може да припаѓаат на една од следниве три категории:

1. Силно поврзани јазли (подмножество од јазли, во кое од еден јазол може да се стигне до било кој друг јазол ако се следи насоката на линковите);
2. Транзитивни јазли (кои имаат само излезни линкови, и до нив не може да се пристигне од било кој линк);
3. Апсорбирачки јазли (кои имаат само влезни линкови, и ако еднаш се стигне до нив не може да се излезе).

2.1.2 Претставување со граф

Графот може графички да се претстави како множество од јазли поврзани со множество од линкови како што е прикажано на Фигура 2.1.



Фигура 2. 1 Графичко претставување.

За опишување на графот може да се користи квадратна матрица од ред $m \times m$ која се нарекува **матрица на соседство**, каде m е бројот на јазлите. Елементите на матриците да ги означиме со e_{ij} , при што e_{ij} ќе прима вредност 1 ако постои линк помеѓу јазлите v_i и v_j , ако не постои линк помеѓу јазлите v_i и v_j тогаш e_{ij} ќе прима вредност 0. За насочен граф, елементот e_{ij} прима вредност 1 ако постои линк од јазолот v_i до јазолот v_j . Матрицата на соседство за ненасочен граф е симетрична т.е $e_{ij} = e_{ji}$. Бидејќи претпоставивме дека графот нема јамки, јасно е дека дијагоналните елементи $e_{ii} = 0$. Во Табела 2.1 е дадена матрицата на соседство на графот претставен на Фигура 2.1.

Табела 2. 1 Матрица на соседство на графот претставен на Фигура 2.1

	s	v_1	v_2	t
s	0	1	1	0
v_1	0	0	1	1
v_2	0	1	0	1
t	0	0	0	0

Графот може исто така да се претстави и со **листа на соседство** каде за секој јазол v_i е дадена листа од соседи. На Табела 2.2 е дадена листата на соседство на графот претставен на Фигура 2.1.

Табела 2. 2 Листа на соседство на графот претставен на Фигура 2.1

јазол	листа
s	$v_1 v_2$
v_1	$v_2 t$
v_2	$v_1 t$
t	

2.2 Надежност на бинарни транспортни системи

При разгледување на бинарни транспортни системи се претпоставува дека компонентите на системот, како и самиот систем може да бидат во една од две можни состојби: да бидат во работна состојба, т.е. исправни или да бидат расипани. Овде ние прво ќе ја дефинираме **структурната функција** на бинарниот систем. Како што веќе е кажано во поглавјето 1.1, структурната функција ни овозможува да го опишеме однесувањето на системот во зависност од неговите компоненти. Претпоставуваме дека

системот се состои од n елементи, e_i , $i = 1, 2, \dots, n$. На секој елементи му е придружена случајна променлива на следниот начин:

$$x_i = \begin{cases} 1, & \text{ако компонентата та } e_i \text{ работи} \\ 0, & \text{ако компонентата та } e_i \text{ не работи} \end{cases}.$$

Случајната променлива x_i претставува состојба во која се наоѓа компонентата e_i . Затоа можеме да дефинираме случаен вектор $\vec{x} = (x_1, x_2, \dots, x_n)$ што претставува вектор од состојби на компонентите на системот. Постојат вкупно вакви 2^n вектори, што значи дека имаме 2^n различни состојби на множеството од компоненти на системот.

Нека y е случајна променлива која ја опишува состојбата на системот. Случајната променлива ја дефинираме на следниот начин:

$$y = \begin{cases} 1, & \text{ако системот работи} \\ 0, & \text{ако системот не работи} \end{cases}$$

Јасно е дека случајната променлива y зависи од случајниот вектор \vec{x} оваа зависност се опишува со помош на **структурата функција**.

$$y = \varphi(\vec{x}) \quad \text{или} \quad y = \varphi(x_1, x_2, \dots, x_n).$$

2.2.1 Анализа на надежност на транспортни системи

Традиционално, техниките кои се користеле за анализа на надежност на системите се применувале на транспортни системи составени од десетици или стотици јазли. Овие техники се базирале на исцрпувачки алгоритми за пребарување со цел да обезбедат квалитативни и квантитативни информации за поврзаност на линковите во транспортниот систем, зависностите што постојат во системот и грешките кои може да се појават на јазлите или линковите на системот [29].

При дизајнирање на транспортните системи, главна грижа е да се дизајнираат системи кои ќе може да функционираат ефикасно и во случај кога некои компоненти на системот се расипани или не работат со полн капацитет. Како прво нè интересира поврзаност на јазлите на системот. Треба да се нагласи дека во реални системи кои се состојат од неколку илјади јазли, многу почесто се појавуваат проблеми во врска со перформансите на системот. Исто така, треба да се нагласи дека, иако проблемите што може да ги предизвикаат јазлите многу поретко се појавуваат, тие може да доведат до катастрофални последици, како на пример расипување на целиот систем. Кога се анализира надежноста на транспортните системи се поставуваат следниве три прашања:

1. Каква е структурата на транспортниот систем.
2. Кои се причините кои може да доведат до грешка на компонентите.
3. Кога се смета дека функционира (е во работна состојба) транспортниот систем.

Било која мерка за надежноста на системот треба да е корисен квантитативен индикатор за способност на системот да функционира и покрај тоа што одредени негови компоненти се целосно или делумно расипани. За да се даде прецизна дефиниција на надежноста потребно е да се даде прецизен одговор на трите погоре поставени прашања. Да се даде одговор на овие прашања и не е толку едноставно, ако се земе во предвид дека се испитуваат различни типови на системи. Исто така, постојат различни причини кои може да доведат до расипување на одредени компоненти, а и одредени системи може да функционираат или не под различни услови (бинарни или повеќе – состојбени системи). Поради севе ова, не може да се очекува да постои единствена дефиниција за надежност на транспортните системи. Затоа подолу ќе се фокусираме на една од попознатите методи за проучување на надежност на транспортни системи.

Да го разгледаме транспортниот систем $G(V, E)$, каде V е множество од јазли, E е множество од линкови. Линковите може да бидат насочени или ненаочени. Исто така во моделите што ќе ги разгледуваме претпоставуваме дека јазлите никогаш нема да се расипат, додека линкови може да се расипуваат. За секој линк $e \in E$, со p_e се означува веројатност дека линкот е во работна состојба. Исто така, да нагласиме дека линковите помеѓу себе се статистички независни, т.е. дали еден линк ќе работи или не зависи од состојбата во која се наоѓаат другите линкови.

Да разгледаме едно основно функционирање на транспортниот систем. Нека $K \subseteq V$, $|K| = k$, е множество од јазли, да претпоставиме дека сите овие јазли комуницираат еден со друг. **k - терминална надежност** е веројатноста дека постои подмножество од k ($2 \leq k \leq |V|$) поврзани јазли во транспортниот систем. Кога $K = V$, тогаш се разгледува поврзанооста помеѓу сите јазли во системот и станува збор за комплетно терминална надежност, R_{all} :

$$R_{all} = P(\text{сите } |V| \text{ јазли се поврзани}).$$

Ако $|K| = 2$, тогаш разгледуваме двотерминална надежност со извор s и сливник t , R_{st} :

$$R_{st} = P(\text{јазлите } s \text{ и } t \text{ се поврзани}).$$

Од интерес на овој труд се транспортните системи кои имаат еден извор и еден сливник, т.е. двотерминалните транспортни системи.

2.2.2 Преглед на методите за пресметување на надежност на бинарни транспортни системи

Методите за пресметување на надежност на транспортни системи, може да ги класифицираме во следните четири категории:

1. Методи со набројување (енумерирачки методи);
2. Методи со трансформирање (редукција и декомпозиција);
3. Директни методи;
4. Апроксимативни методи.

Значајно е да се нагласи дека овие методи не се исклучуваат заемно и често се користат нивни комбинации. Методот со набројување е наједноставен но најнеефикасен метод (бројот на сите можни состојби вообичаено расте експоненцијално со зголемување на транспортниот систем). Многу поефикасни се методите со набројување на минималните пат и минималните пресек множества, што резултираат во соодветна DNF(дисјунктна нормална форма) претставување на структурната функција. Меѓутоа и овие методи многу често може да биде неизводливи во пракса. Дали претставувањето на структурната функција преку минимални патишта или минимални пресеци е погодно, зависи од тоа каква е структурата на транспортниот систем што се разгледува. Од неодамна се воведени и различни пристапи кои се базираат на дијаграмот на бинарно одлучување (Binary Decision Diagrams – BDD) каде структурната функција се претставува со помош на методот на декомпозиција на Shannon. Блиски до овие пристапи се пристапите кои се базираат на дијаграмите со проширени линкови (Edge Expansion Diagrams – EEDs).

Друг метод што често се користи за пресметување на надежноста е со помош на стандардната формула за вклучување и исклучување. Да разгледаме специјален случај во кој се претпоставува дека надежноста на сите линкови на мрежата е еднаква. Нека p е веројатноста некој линк да се појавува во пат множество, тогаш веројатноста дека во пат множеството ќе се појават i линкови е $p^i(1-p)^{n-i}$, каде n е бројот на линковите во транспортниот систем. Во овој случај надежноста може да се напише на следниот начин :

$$R(p) = \sum_{i=0}^n N_i p^i (1-p)^{n-i}, \quad (2.1)$$

каде N_i е бројот на пат множествата што се составени од i линкови [9], [10], [23], [35].

Покрај сите горенаведени методи, постојат и други различни методи со декомпозиција (разложување) и редуцирање. Овие методи може да се користат независно од горе спомнатите методи. Главна идеја на методите со декомпозиција (разложување) е да се подели графот на подграфови, и да се оценат (да се пресмета нивната надежност) подграфовите независно еден од друг. Надежноста на целиот граф потоа се пресметува врз основа на надежностите што се добиени за соодветните подграфови. Техниките со редуцирање се специјални техники за трансформација кои овозможуваат да се поедностави тополошката структура на дадената мрежа. Целта на оваа техника е да се намали големината на мрежата, со цел да се подобрат перформансите за идни пресметувања на надежноста. За некои класи на графови, надежноста може да се пресмета ефикасно со само една примена на методот со редуцирање [11].

Покрај методите за точно пресметување на надежноста постојат апроксимативни техники за пресметување на надежноста. Тоа се должи на фактот што пресметување на надежноста на мрежите е NP – тежок проблем. Исто така со помош на техниките на Монте – Карло се развиваат многу софистицирани методи и за оценување на надежноста на системи и за одредување на границите на надежноста.

Во следните подглави накратко ќе ги опишеме методите кои се применуваат при пресметување на надежноста на транспортни системи. Нивното значење се состои во што што тие претставуваат основа за пишување на алгоритми за пресметување на надежноста на транспортни системи.

2.2.2.1 Метод со набројување на сите можни состојби

Методот со набројување на сите можни состојби (или метод со таблица на вистинитости вредности) се базира на идејата да се определат сите можни комбинации од состојби на линковите на системот.

Бидејќи мрежата има 5 линкови, бројот на сите можни комбинации од состојбите на линковите е $2^5=32$. Надежноста на транспортниот систем ќе ја пресметаме како сума од веројатностите на состојбите со кои транспортниот систем е во работна состојба, т.е. за кои постои пат од изворот s до сливникот t . Нека $\vec{x} = (x_1, x_2, x_3, x_4, x_5)$ е вектор од состојби на линковите на транспортниот систем од Фигура 2.1. Нека p_i , е веројатноста дека i – от линк е во работна состојба т.е. $p_i = P\{x_i = 1\}$, $i = 1, 2, \dots, 5$, додека $q_i = (1 - p_i) = P\{x_i = 0\}$, е веројатноста дека линкот не е во работна состојба, $i = 1, 2, \dots, 5$. Во Табела 2.3 се дадени сите комбинации од состојби на линкови за кои системот е во работна состојба. Надежноста на системот се пресметува како сума од веројатностите дадени во втората колона од табелата.

Табела 2.3 Вектор од состојби во кои транспортниот систем од Фигура 2.1 е во работна состојба

$\vec{x} = (x_1, x_2, x_3, x_4, x_5)$	Веројатност
(1, 1, 0, 0, 0)	$p_1 p_2 q_3 q_4 q_5$
(0, 0, 0, 1, 1)	$q_1 q_2 q_3 p_4 p_5$
(1, 1, 1, 0, 0)	$p_1 p_2 p_3 q_4 q_5$
(1, 1, 0, 1, 0)	$p_1 p_2 q_3 p_4 q_5$
(1, 1, 0, 0, 1)	$p_1 p_2 q_3 q_4 p_5$
(1, 0, 1, 0, 1)	$p_1 q_2 p_3 q_4 p_5$
(1, 0, 0, 1, 1)	$p_1 q_2 q_3 p_4 p_5$
(0, 1, 1, 1, 0)	$q_1 p_2 p_3 p_4 q_5$
(0, 1, 0, 1, 1)	$q_1 p_2 q_3 p_4 p_5$
(0, 0, 1, 1, 1)	$q_1 q_2 p_3 p_4 p_5$
(1, 1, 1, 1, 0)	$p_1 p_2 p_3 p_4 q_5$
(1, 1, 1, 0, 1)	$p_1 p_2 p_3 q_4 p_5$
(1, 1, 0, 1, 1)	$p_1 p_2 q_3 p_4 p_5$
(1, 0, 1, 1, 1)	$p_1 q_2 p_3 p_4 p_5$
(0, 1, 1, 1, 1)	$q_1 p_2 p_3 p_4 p_5$
(1, 1, 1, 1, 1)	$p_1 p_2 p_3 p_4 p_5$

2.2.2.2 Метод на декомпозиција (факторизација) (Factoring method)

Методот на декомпозиција се фокусира на состојбите на поединечните линкови. Овој метод се базира на теоремата за факторизација:

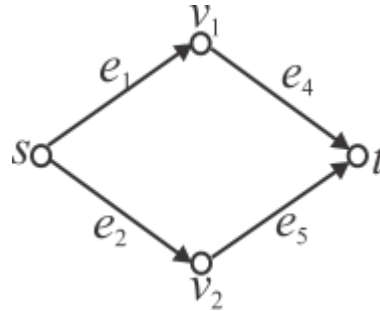
$$R(G) = p_e R(G/\text{линкот } e \text{ е во работна состојба}) + (1 - p_e) R(G/\text{линкот } e \text{ не е во работна состојба}). \quad (2.2)$$

каде p_e е веројатноста дека линкот e е во работна состојба. Овој метод може да се примени на било кој линк e на графот G . Идејата е да се избере одреден линк од графот и да се разгледуваат два случаја:

- кога соодветниот линк е во работна состојба,
- кога соодветниот линк не е во работна состојба.

За секој случај се добиваат два нови графови, едниот под претпоставка дека линкот е во работна состојба, другиот под претпоставка дека линкот не е во работна состојба.

Да го разгледаме методот на графот прикажан на Фигура 2.1. Ако го разгледаме линкот e_3 ќе ги добиеме графовите претставени на Фигура 2.2 и Фигура 2.3.

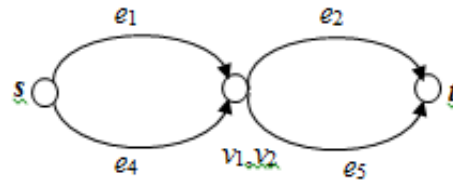


Фигура 2.2 Линкот e_3 не е во работна состојба.

Нека p_i е веројатноста дека линкот e_i е во работна состојба, т.е. $p_i = P\{e_i = 1\}$. За транспортниот систем од Фигурата 2.2 пат од s до t ќе постои ако се во работна состојба линковите e_1 и e_4 , или линковите e_2 и e_5 . Па надежноста можеме да ја пресметаме како:

$$R(G/\text{линкот } e_3 \text{ не е во работна состојба}) = 1 - (1 - p_1 p_2) (1 - p_4 p_5), \quad (2.3)$$

Ако претпоставиме дека линкот e_3 е во работна состојба, тогаш можеме да ги споиме јазлите v_1 и v_2 во еден јазол и го добиваме графот претставен на Фигура 2.3.



Фигура 2.3 Линкот e_3 е во работна состојба.

Графот претставен на Фигурата 2.3 можеме да го поделиме на два подграфа, во кој првиот подграф се состои од линкови e_1 и e_4 (за да е во работна состојба овој подграфот барем еден од линковите e_1 или e_4 треба да е во работна состојба), додека вториот подграф се состои од линковите e_2 и e_5 (за да е во работна состојба подграфот барем еден од линковите e_2 или e_5 треба да е во работна состојба).

$$\begin{aligned} R(G/\text{линкот } e_3 \text{ е во работна состојба}) &= (1 - (1 - p_1)(1 - p_4)) (1 - (1 - p_2)(1 - p_5)) \\ &= (1 - q_1 q_4) (1 - q_2 q_5), \end{aligned} \quad (2.4)$$

каде $q_i = 1 - p_i = P\{e_i = 0\}$.

Ако ги применеме равенствата (2.3) и (2.4) во (2.2) за надежноста на транспортниот систем од Фигура 2.1 се добива.

$$R = p_{e_3} R(G/\text{линкот } e_3 \text{ е во работна состојба}) + q_{e_3} R(G/\text{линкот } e_3 \text{ не е во работна состојба}) \quad (2.5)$$

2.2.2.3 Метод со примена на формула за вклучување и исклучување

Во поглавјето 1.1 ги дадовме дефинициите за пат, минимален пат, пресек и минимален пресек. Исто така дадовме еден начин за пресметување на надежноста со формулата за вклучување и исклучување ако минималните патишта или ако минималните пресеци се познати. Овде повторно ќе ги дадеме истите формули, но и со нивна помош ќе ја пресметаме надежноста на транспортниот систем даден на Фигура 2.1. Да претпоставиме дека P_1, P_2, \dots, P_l се минимални патишта, тогаш надежноста е еднаква на веројатноста од унијата на l – те минимални патишта, т.е.

$$R = P\left(\bigcup_{h=1}^l P_h\right), \quad (2.6)$$

Со примена на формулата за вклучување и исклучување се добива

$$R = P\left(\bigcup_{h=1}^l P_h\right) = \sum_{h=1}^l P(P_h) - \sum_{h < k} P(P_h \cap P_k) + \dots + (-1)^l P(P_1 \cap \dots \cap P_l). \quad (2.7)$$

каде l е бројот од минимални патишта и P_h е минималниот пат h .

Додека ако C_1, C_2, \dots, C_w се минимални пресеци, тогаш надежност може да се пресмета со следната формула:

$$R = 1 - P\left(\bigcup_{h=1}^w C_h\right) = 1 - \left(\sum_{h=1}^w P(C_h) - \sum_{h < k} P(C_h \cap C_k) + \dots + (-1)^w P(C_1 \cap \dots \cap C_w)\right). \quad (2.8)$$

каде w е бројот од минимални пресеци и C_h е минималниот пресек h .

Со S_k да ја означиме сумата од веројатностите на пресеците на било кои k минимални патишта, т.е.

$$S_k = \sum_{1 \leq i_1 < \dots < i_k \leq l} P(P_{i_1} \cap \dots \cap P_{i_k}), \quad (2.9)$$

тогаш со примена на формулата за вклучување и исклучување, надежноста на системот, која е еднаква на веројатноста од унијата на l минимални патишта, може да биде изразена како:

$$R = \sum_{k=1}^l (-1)^{k-1} \mathcal{S}_k. \quad (2.10)$$

Да ја пресметаме надежноста на транспортниот систем на Фигура 2.1 со примена на формулата за вклучување и исклучување врз минималните патишта. Нека $p_i = P\{e_i = 1\} = P(e_i)$ и $q_i = P\{e_i = 0\} = P(\bar{e}_i)$. Бидејќи се претпоставува дека линковите се независни, $P(e_i e_j) = P(e_i)P(e_j)$, за $i \neq j$. Минимални патишта за транспортниот систем од Фигура 2.1 се $\mathcal{P}_1 = \{e_1, e_2\}$, $\mathcal{P}_2 = \{e_4, e_5\}$, $\mathcal{P}_3 = \{e_1, e_3, e_5\}$ и $\mathcal{P}_4 = \{e_2, e_3, e_4\}$.

$$\begin{aligned} \mathcal{S}_1 &= P(\mathcal{P}_1) + P(\mathcal{P}_2) + P(\mathcal{P}_3) + P(\mathcal{P}_4), \\ \mathcal{S}_2 &= P(\mathcal{P}_1 \mathcal{P}_2) + P(\mathcal{P}_1 \mathcal{P}_3) + P(\mathcal{P}_1 \mathcal{P}_4) + P(\mathcal{P}_2 \mathcal{P}_3) + P(\mathcal{P}_2 \mathcal{P}_4) + P(\mathcal{P}_3 \mathcal{P}_4), \\ \mathcal{S}_3 &= P(\mathcal{P}_1 \mathcal{P}_2 \mathcal{P}_3) + P(\mathcal{P}_1 \mathcal{P}_2 \mathcal{P}_4) + P(\mathcal{P}_1 \mathcal{P}_3 \mathcal{P}_4) + P(\mathcal{P}_2 \mathcal{P}_3 \mathcal{P}_4), \\ \mathcal{S}_4 &= P(\mathcal{P}_1 \mathcal{P}_2 \mathcal{P}_3 \mathcal{P}_4). \\ R &= \mathcal{S}_1 - \mathcal{S}_2 + \mathcal{S}_3 - \mathcal{S}_4, \\ R &= P(e_1 e_2) + P(e_4 e_5) + P(e_1 e_3 e_5) + P(e_2 e_3 e_4) - P(e_1 e_2 e_4 e_5) - P(e_1 e_2 e_3 e_5) - \\ &\quad P(e_1 e_2 e_3 e_4) - P(e_1 e_3 e_4 e_5) - P(e_2 e_3 e_4 e_5) - P(e_1 e_2 e_3 e_4 e_5) + P(e_1 e_2 e_3 e_4 e_5) + \\ &\quad P(e_1 e_2 e_3 e_4 e_5) + P(e_1 e_2 e_3 e_4 e_5) - P(e_1 e_2 e_3 e_4 e_5), \\ R &= p_1 p_2 + p_4 p_5 + p_1 p_3 p_5 + p_2 p_3 p_4 - p_1 p_2 p_4 p_5 - p_1 p_2 p_3 p_5 - p_1 p_2 p_3 p_4 - p_1 p_3 p_4 p_5 \\ &\quad - p_2 p_3 p_4 p_5 + 2 p_1 p_2 p_3 p_4 p_5 \end{aligned} \quad (2.11)$$

2.2.2.4 Метод со сума од дисјунктни производи (SDP)

Со E_i да го означиме настанот, сите линкови од минималниот пат \mathcal{P}_i се во работна состојба. Тогаш надежноста може да се пресмета со следниот израз:

$$R = P(E_1) + P(\bar{E}_1 E_2) + \dots P(\bar{E}_1 \bar{E}_2 \dots \bar{E}_{l-1} E_l). \quad (2.12)$$

каде l е бројот на минимални патишта, и \bar{E}_i е комплемент на настанот E_i .

Со цел да ја примениме техниката SDP, треба да ја пресметаме веројатноста $P(\sigma)$, каде $\sigma_i = \bar{E}_1 \bar{E}_2 \dots \bar{E}_{l-1} E_l$. Ако ги дефинираме настаните σ_i во термини на минимални патишта, тогаш σ_i го претставува настанот, сите компоненти од минималниот пат \mathcal{P}_i се во работна состојба, и секој од минималните патишта $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_{i-1}$ содржи барем една компонента што не е во работна состојба. Ако E_i и E_j се независни за $1 \leq j \leq i$, тогаш

$$P(\sigma_i) = \prod_{k \in \mathcal{P}_i} p_k \prod_{j=1}^{i-1} [1 - \prod_{s \in \mathcal{P}_j} p_s]. \quad (2.13)$$

каде p_k е веројатноста дека k -от линк е во работна состојба.

Ако минималните патишта имаат заеднички линкови, тогаш $P(\sigma_i)$ се пресметува на следниот начин:

$$P(\sigma_i) = P(E_i) P(\bar{E}_1 \setminus E_i) P(\bar{E}_2 \setminus E_i E_1) \dots P(\bar{E}_{i-1} \setminus E_i E_1 E_2 \dots E_{i-2}). \quad (2.14)$$

Нека \mathcal{T}_j е условниот настан дека минималниот пат \mathcal{P}_j не е во работна состојба при услов дека минималните патишта $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_{j-1}$ се во работна состојба. Тогаш добиваме:

$$P(\sigma_i) = P(E_j) \prod_{j=1}^{i-1} P(\mathcal{T}_j). \quad (2.15)$$

$P(E_i)$ ја претставува веројатноста на настанот E_i и оваа веројатност лесно може да се пресмета ако се претпостави дека расипувањата на линковите се независни. Во спротивно, доколку постои зависност помеѓу расипувањата, пресметувањето на $P(\mathcal{T}_j)$ е комплексен проблем.

Нека повторно $p_i = P\{e_i = 1\}$ и $q_i = P\{e_i = 0\}$. За транспортниот систем претставена на Фигура 2.1, со користење на минималните патишта за надежност на транспортниот систем се добива:

$$\begin{aligned} R &= P(e_1 e_2) + P(\overline{e_1} e_2 e_4 e_5) + P(\overline{e_1} e_2 e_4 e_5 e_1 e_3 e_5) + P(\overline{e_1} e_2 e_4 e_5 e_1 e_3 e_5 e_2 e_3 e_4) \\ &= p_1 p_2 + q_1 p_4 p_5 + q_2 p_4 p_5 + q_2 q_4 p_1 p_3 p_5 + q_1 q_5 p_2 p_3 p_4. \end{aligned} \quad (2.16)$$

Ако го замениме q_i со $1 - p_i$, ќе го добиеме истиот резултат што го користење на формулата за вклучување и исклучување со равенката (2.11).

2.3 Определување на минимални пресек множества во бинарен транспортен систем

Како што истакнавме и во воведниот дел, еден од најкомплексните проблеми во анализата на надежност на системи е наоѓањето на минималните пресеци и патишта. Минималните пресеци најчесто се користат кај високо надежни системи, бидејќи со овој пристап се добиваат мали грешки при заокружување. Проблемот за развој на алгоритми со кои ќе се добијат минималните пресек множества е еден од најчесто анализираните проблеми. Вообичаено се разгледуваат насочени и ациклични транспортни системи. Алгоритмот предложен во [1] е базиран на пристапот на замена на множества и стратегијата на замена на елементи со кои се добиваат минималните пресек множества поврзани со даден транспортен систем. Алгоритмот што е предложен со овој труд се разликува од другите алгоритми во литературата од тој период, и тој се базира на матрица на претходници и техники за замена на елементи што овозможуваат точно пресметување на минималните пресек множества и непосредно вклучување на грешките на јазлите без да се врши каква било промена на псевдокодот.

Во алгоритмот предложен во [5] проблемот на енумерација на пресек множествата меѓу сите парови од јазли во графот се сведува на решавање на систем на линеарни равенки, при што е предложена алгебра за пресек множествата. Исто така е предложен и нов алгоритам за решавање на системот линеарни равенки со користење на гаусова елиминација. Бидејќи и едноставна имплементација на собирање и множење во предложената алгебра може да го направи алгоритмот на Гаус доста неефикасен, покажано е како овие операции може да бидат имплементирани на ефикасен начин за

ненасочен граф, ако се искористат некои својства на пресек множествата. Исто така во трудот се пресметува и сложеноста на предложениот алгоритам.

Во многу од предложените алгоритми, се добиваат кандидати за минимални пресек множества што не се минимални. Овие множества може да се елиминираат со некоја додатна процедура што вклучува споредување меѓу сите парови од кандидати за минимални пресек множества. Ова е скапа процедура без разлика на тоа дали бројот на кандидати за минимални пресек множества е голем или не.

Ние предлагаме два пристапи [37] и [39] за определување на минималните пресек множества. Со првиот пристап, директно се наоѓаат сите минимални пресек множества. Со основната верзија на алгоритмот се добиваат и кандидати за минимални пресек множества што не се минимални. За таа цел се дадени две оптимизации на алгоритмот. Со првата оптимизација исти кандидати за минимални пресек множества не се добиваат, но може да се добијат кандидати за минимални пресек множества што не се минимални. Поради тоа направена е оптимизација и на вториот алгоритам. Со цел да се отстранат појавувањата на кандидати за минимални пресек множества што не се минимални, анализираме како пресек множествата се добиени. Ако соодветното множество го дели транспортниот систем на повеќе од две сврзани компоненти, тогаш тоа не е минимално пресек множество.

Со вториот алгоритам што е предложен, користиме BFS пребарување во транспортниот систем. Со овој алгоритам се добиваат само минималните пресек множества. Всушност, предложена е техника за определување на тоа дали определено пресек множество е минимално пресек множество. Уште повеќе, даден е и доказ, дека со предложениот алгоритам се добиваат сите минимални пресек множества. Исто така, анализирана е сложеноста на алгоритмот.

2.3.1 Алгоритми за определување на минимални пресек множества на транспортни системи

Во ова подглавје ќе го разгледаме алгоритмите предложени во [37]. Разгледуваме ненасочен двотерминален транспортен систем $G(V, E)$ каде V е множеството од јазли а E е множеството од линкови, јазолот s е извор, а јазолот t сливник. Како што кажавме и во претходните подглавја, пресек множеството се дефинира како множество од линкови, така што ако нема проток низ овие линкови ќе нема проток и од изворот до сливникот. Пресек множеството C е минимално ако не постои друго пресек множество C' , така што $C' \subseteq C$.

За ненасочен транспортен систем следново тврдење е јасно:

Тврдење 2.1 Нека $G(V, E)$ е ненасочен двотерминален поврзан транспортен систем со извор s и сливник t . Тогаш C е пресек множество ако и само ако со бришење на линковите од C , графот $G(V, E)$ е поделен на два подграфа $G(V_1, E_1)$ и $G(V_2, E_2)$, така што $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, изворот $s \in V_1$ и сливникот $t \in V_2$.

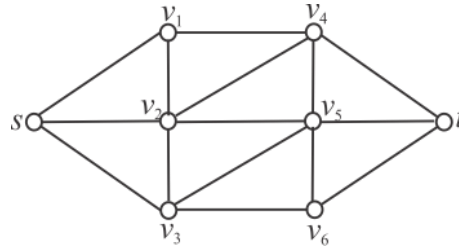
Од наш интерес е да ги определиме минималните пресек множества. Следните тврдење се однесуваат за минималните пресек множества.

Тврдење 2.2 Нека $G(V, E)$ е ненасочен двотерминален поврзан транспортен систем со извор s и сливник t . Ако графот $G(V, E)$ е поделен на два подграфа $G(V_1, E_1)$ и $G(V_2, E_2)$,

така што $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, изворот $s \in V_1$ и сливникот $t \in V_2$, тогаш $C = E / (E_1 \cup E_2)$ е минимално пресек множество.

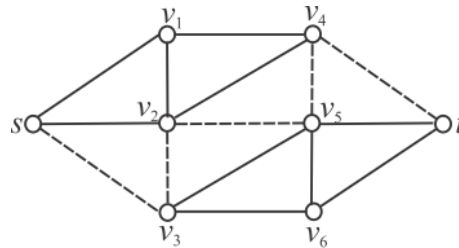
Доказ: Од Тврдењето 2.1, е јасно е дека C е пресек множество. Ние треба да докажеме дека C е минимално пресек множество. Да претпоставиме спротивно, нека C не е минимално пресек множество. Тоа значи дека постои пресек множество C' , така што $C' \subset C$. Исто така е јасно дека $C = \{\{u, v\} \mid u \in V_1 \text{ и } v \in V_2\}$. Треба да покажеме дека постои пат од s до t во графот $G' (V, E / C')$. Бидејќи $C' \subset C$ постои линк $\{u, v\} \in E / C'$ така што $u \in V_1$ и $v \in V_2$. Бидејќи множеството V_1 е поврзано (од секој јазол може да стигнеме до секој друг јазол), тогаш постои пат \mathcal{P}_1 од s до u , а бидејќи и V_2 е поврзано постои пат \mathcal{P}_2 од v до t . Значи патот $\mathcal{P}_1(u, v)\mathcal{P}_2$ е пат од s до t , што е контрадикција со претпоставката дека C е пресек множество. \square

Пример 2.1 Да го разгледаме транспортниот систем прикажан во Фигура 2.4



Фигура 2.4 Ненасочен двотерминален транспортен систем.

На следната Фигура 2.5 е прикажано минималното пресек множество $C = \{\{s, v_3\}, \{v_3, v_2\}, \{v_2, v_5\}, \{v_5, v_4\}, \{v_4, t\}\}$, линковите што се елементи на C се прикажани со испрекинати линии.



Фигура 2.5 Ненасочен двотерминален транспортен систем со пресек множество $C = \{\{s, v_3\}, \{v_3, v_2\}, \{v_2, v_5\}, \{v_5, v_4\}, \{v_4, t\}\}$.

Сега ќе ја предложиме основната верзија на алгоритам за добивање на минимални пресек множества. Со овој алгоритам се добиваат кандидати за минимални пресек множества што не се минимални, и исти минимални пресек множества се добиваат повеќе пати. Главната идеја за развој на овие алгоритми е да се добијат само минималните пресек множества и да не се добива повеќе од еднаш едно минимално пресек множество, Затоа подоцна ќе предложиме и две модификации на основниот алгоритам.

Алгоритам 2.1 – Основна верзија на алгоритам за пресметување на множество од минимални пресеци ($G(V, E)$)

Чекор 1. Се иницијализира множеството A ,

$$A = \{ \{ \{s, v_i\} \mid \{s, v_i\} \in E \}, \{v_i \mid \{s, v_i\} \in E \}, \{s\} \},$$

Чекор 2. Се иницијализира множеството CutSet , $\text{CutSet} = \{ \{s, v_i\} \mid \{s, v_i\} \in E \}$,

Чекор 3. Додека $A \neq \emptyset$, се зема елемент од множеството A , $B = \{B_1, B_2, B_3\}$ и овој елемент се брише од A ,

Чекор 3.1. За секој елемент $v_j \in B_2$, $v_j \neq t$, се наоѓаат следниве множества

$$B_3 = B_3 \cup \{v_j\},$$

$$B_2 = \{B_2 / \{v_j\}\} \cup \{v_i \mid \{v_i, v_j\} \notin B_1\},$$

$$B_1 = \{B_1 / \{\{v_i, v_j\} \mid \{v_i, v_j\} \in B_1\}\} \cup \{\{v_i, v_j\} \mid \{v_i, v_j\} \notin B_1\},$$

$$\text{CutSet} = \text{CutSet} \cup B_1,$$

$$A = A \cup \{\{B_1, B_2, B_3\}\},$$

Чекор 4. Се бришат сите елементи $Y \in \text{CutSet}$, за кои што постои множество X , такво што $X \in \text{CutSet}$ и $X \subset Y$,

Чекор 5. Се враќа CutSet .

Лема 2.1 Во секоја итерација (повторување) во Чекор 3.1 од Алгоритам 2.1 графовите $G_1(B_3, \{\{u, v\} \mid u, v \in B_3\})$ и $G_2(V/B_3, \{\{u, v\} \mid u, v \in V/B_3\})$ го делат графот $G(V, E)$ во две несврзани компоненти, така што изворот s припаѓа во G_1 и сливникот t во G_2 .

Доказ: Јасно е дека $B_3 \cup V/B_3 = V$ и $B_3 \cap V/B_3 = \emptyset$, што значи дека графовите G_1 и G_2 го делат графот G на две несврзани компоненти. Бидејќи во Чекор 1, го иницијализиравме $B_3 = \{s\}$ и со секој влез во Чекор 3.1 ние додаваме само еден елемент во множеството B_2 , следува дека $s \in G_1$. Од друга страна никогаш не го додаваме сливникот во B_3 , т.е. $t \notin B_3$. Од каде што следува дека s и t припаѓаат во различни компоненти. \square

Теорема 2.1 За секој граф $G(V, E)$ со извор s и сливник t што е влез на Алгоритмот 2.1, елементите од множеството CutSet се пресек множества.

Доказ: Иницијално имаме дека множеството $\text{CutSet} = \{B_1\}$, $B_1 = \{\{s, v_i\} \mid \{s, v_i\} \in E\}$, што значи дека пред да влезе во циклусот во Чекор 3.1, CutSet се состои од пресек множества за графот G . Ние ќе покажеме дека тврдењето “ B_1 е пресек множество” е инваријанта (непроменливо) на циклусот во Чекор 3.1. Всушност треба да покажеме дека (2.17)

$$B_1 = \{\{u, v\} \mid u \in B_3, v \in V/B_3\} \quad (2.17)$$

е инваријанта на циклусот во Чекор 4.1.

Претпоставуваме дека пред да влезе во циклусот $B_1 = \{\{u, v\} \mid u \in B_3, v \in V/B_3\}$ е пресек множество. Нека $B_3' = B_3 \cup \{v_j\}$, $B_2' = \{B_2 / \{v_j\}\} \cup \{v_i \mid \{v_i, v_j\} \notin B_1\}$ и $B_1' = \{B_1 / \{\{v_i, v_j\} \mid \{v_i, v_j\} \in B_1\}\} \cup \{\{v_i, v_j\} \mid \{v_i, v_j\} \notin B_1\}$. Од Лема 2.1 следува дека графовите $G_1(B_3', \{\{u, v\} \mid u, v \in B_3'\})$ и $G_2(V/B_3', \{\{u, v\} \mid u, v \in V/B_3'\})$ го делат графот G на две компоненти така

што $s \in B_3'$ и $t \in V/B_3'$. За да покажеме дека B_1' е пресек множество, ние треба да покажеме дека $B_1' = \{\{u, v\} \mid u \in B_3', v \in V/B_3'\}$, т.е.

$$\{(u, v) \mid u \in B_3', v \in V/B_3'\} = \{B_1 / \{(v_i, v_j) \mid (v_i, v_j) \in B_1\}\} \cup \{(v_i, v_j) \mid (v_i, v_j) \notin B_1\} \quad (2.18)$$

Нека $\{u', v'\} \in \{\{u, v\} \mid u \in B_3', v \in V/B_3'\} \Leftrightarrow u' \in B_3' \wedge v' \in V/B_3' \Leftrightarrow u' \in B_3 \cup \{v_j\} \wedge v' \in V/\{B_3 \cup \{v_j\}\} \Leftrightarrow (u' \in B_3 \vee u' \in \{v_j\}) \wedge v' \in V/\{B_3 \cup \{v_j\}\} \Leftrightarrow (u' \in B_3 \wedge v' \in V/\{B_3 \cup \{v_j\}\}) \vee (u' = v_j \wedge v' \in V/\{B_3 \cup \{v_j\}\})$

I случај: $u' \in B_3 \wedge v' \in V/\{B_3 \cup \{v_j\}\} \Leftrightarrow u' \in B_3 \wedge v' \notin B_3 \cup \{v_j\} \Leftrightarrow u' \in B_3 \wedge \neg (v' \in B_3 \vee v' = v_j) \Leftrightarrow u' \in B_3 \wedge v' \in V/B_3 \wedge v' \neq v_j \Leftrightarrow (u' \in B_3 \wedge v' \in V/B_3) \wedge (u' \in B_3 \wedge v' \neq v_j) \Leftrightarrow \{u', v'\} \in B_1 \wedge \{u', v'\} \notin \{\{v_i, v_j\} \mid \{v_i, v_j\} \in B_1\} \Leftrightarrow \{u', v'\} \in \{B_1 / \{\{v_i, v_j\} \mid \{v_i, v_j\} \in B_1\}\}$

II случај: $u' = v_j \wedge v' \in V/\{B_3 \cup \{v_j\}\} \Leftrightarrow u' = v_j \wedge v' \in V/B_3 \wedge v' \neq v_j \Leftrightarrow u', v' \in V/B_3 \wedge v' \neq v_j \wedge u' = v_j$. Бидејќи $B_1 = \{\{u, v\} \mid u \in B_3, v \in V/B_3\}$, $u', v' \in V/B_3$, ако и само ако $\{u', v'\} \notin B_1$ т.е. $\{u', v'\} \in \{\{v_i, v_j\} \mid \{v_i, v_j\} \notin B_1\}$.

Од I случај и II следува дека (2.18) е вистинито. \square

За да се докаже дека со Алгоритам 2.1 се добиваат сите минимални пресеци, потребно е следново тврдење.

Тврдење 2.3 Нека $G(V, E)$ е сврзан граф со $|V| > 1$. Тогаш постојат најмалку два јазли $x, y \in V$, за кои $G(V/\{x\}, E/\{\{x, u\} \mid \{x, u\} \in E\})$ и $G(V/\{x\}, E/\{\{y, u\} \mid \{y, u\} \in E\})$ се сврзани графови.

Теорема 2.2 Со Алгоритмот 2.1 се добиваат сите минимални пресек множества за графот $G(V, E)$ што е влез на алгоритмот.

Доказ: Прво треба да покажеме дека секое множество од јазли B_3 , за кое важи дека $s \in B_3$ и $G_1(B_3, \{\{u, v\} \mid u, v \in B_3\})$ е сврзан граф, може да се добие со предложениот алгоритам. Јасно е дека единствено вакво множество со еден јазол е множеството $\{s\}$, $G_1(\{s\}, \emptyset)$. Да претпоставиме дека на овој начин се добиени сите множества B_3 , $|B_3| = k$, така што $s \in B_3$ и $G_1(B_3, \{\{u, v\} \mid u, v \in B_3\})$ е сврзан граф, за $k = 1, 2, \dots, n - 1$.

Нека Y е множество од јазли од G , така што $|Y| = n$, $s \in Y$ и графот $G_1'(Y, \{\{u, v\} \mid u, v \in Y\})$ е сврзан. Од Тврдењето 2.3 следува дека постои јазол v_j , $v_j \neq s$, така што графот $G_1''(Y', \{\{u, v\} \mid u, v \in Y'\})$, каде $Y' = Y \cup \{v_j\}$ е сврзан граф и $s \in Y'$. Бидејќи $|Y'| = n - 1$ од индуктивната претпоставка имаме дека Y' е добиено со Алгоритам 2.1. Бидејќи постои линк $\{u, v_j\} \in E$ и $u \in Y'$ јасно е дека и $Y = Y' \cup \{v_j\}$ е добиено со Алгоритам 2.1.

На крај да заклучиме, секое минимално пат множество го дели графот на две сврзани компоненти и сите вакви компоненти се добиваат со Алгоритам 2.1. Значи со Алгоритам 2.1 се добиваат сите минимални пресек множества. \square

Недостаток на овој алгоритам е што може да се добијат исти пресек множества повеќе пати. Од овие причини, ние ќе предложиме оптимизација на претходниот алгоритам. За таа цел, разгледуваме и друго множество F . Во F ќе ги чуваме сите јазли u_i

$\in B_2$, за кои може да најдеме јазли u_j , така што $\{u_i, u_j\} \in E$ и $\{u_i, u_j\}$ не е линк во ниту едно од претходно добиените пресек множества. Со цел да се избегне појавување на дупликати за истата комбинација од јазли во секоја наредна итерација, ние ќе ги избришеме овие јазли од B_2 . На овој начин ќе добиеме подобрување на алгоритмот, со тоа што третата компонента во множеството A се добива само еднаш, и поради тоа секој елемент од A е добиен само еднаш, т.е. секое пресек множество е добиено само еднаш.

Модифицираниот алгоритам е:

Алгоритам 2.2 – Модифициран алгоритам за пресметување на множество од минимални пресеци ($G(V, E)$)

Чекор 1. Се иницијализира множеството A , $A = \{\{\{s, v_i\} \mid \{s, v_i\} \in E\}, \{v_i \mid \{s, v_i\} \in E\}\}$,

Чекор 2. Се иницијализира множеството CutSet , $\text{CutSet} = \{\{s, v_i\} \mid \{s, v_i\} \in E\}$,

Чекор 3. Додека $A \neq \emptyset$, извршувај ги чекорите од 3.1 до 3.3.1.4,

Чекор 3.1 $B = A[1]$, $A = A/B$,

Чекор 3.2 $Z = B[2]$; $n = \text{Length}(Z)$; $F = \emptyset$,

Чекор 3.3 За $i = 1$ до n

Чекор 3.3.1 Ако $Z[i] \neq t$ тогаш

Чекор 3.3.1.1 $v_j = Z[i]$; $F = F \cup \{v_j\}$,

Чекор 3.3.1.2 $D = \{B[1] / \{\{v_i, v_j\} \mid \{v_i, v_j\} \in B[1]\}\} \cup \{\{v_i, v_j\} \mid \{v_i, v_j\} \notin B[1]\}$, $\text{CutSet} = \text{CutSet} \cup D$,

Чекор 3.3.1.3 $H = \{Z \cup \{v_i \mid \{v_i, v_j\} \notin B[1]\}\} / F$,

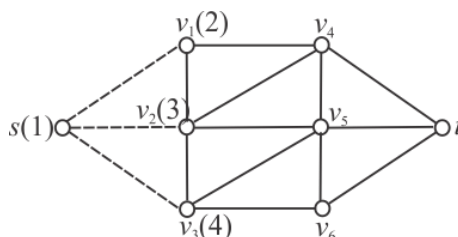
Чекор 3.3.1.4 $A = A \cup \{D, H\}$,

Чекор 4. Се бришат сите $Y \in \text{CutSet}$, за кои што постои множество X , такво што $X \in \text{CutSet}$ и $X \subset Y$.

Чекор 5. Се враќа CutSet .

Пример 2.2 Преку овој пример ќе покажеме како работи Алгоритам 2.2. Повторно ќе ја разгледаме мрежата од Фигура 2.4.

Со примена на Чекор 1, се добива $A = \{\{\{s, v_1\}, \{s, v_2\}, \{s, v_3\}\}, \{v_1, v_2, v_3\}\}$ и $\text{CutSet} = \{\{s, v_1\}, \{s, v_2\}, \{s, v_3\}\}$. Од множеството A го наоѓаме множеството $B = \{\{\{s, v_1\}, \{s, v_2\}, \{s, v_3\}\}, \{v_1, v_2, v_3\}\}$ и новото множество A е $A = A/B = \emptyset$. На Фигура 2.6 е прикажано пресек множеството $\{s, v_1\}, \{s, v_2\}, \{s, v_3\}$:



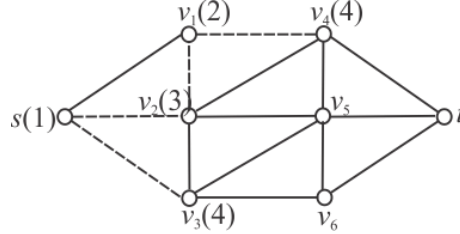
Фигура 2.6 Ненасочен двотерминален транспортен систем со пресек множество $C = \{\{s, v_1\}, \{s, v_2\}, \{s, v_3\}\}$.

Од овде $Z = \{v_1, v_2, v_3\}$, $n = 3$ и $F = \emptyset$.

За секој елемент од Z ќе добиеме различно пресек множество.

$$\begin{aligned}
 Z[1] &= v_1, F = \{v_1\}, \\
 D &= \{\{s, v_2\}, \{s, v_3\}, \{v_1, v_2\}, \{v_1, v_4\}\}, \text{ -- пресек множество} \\
 H &= \{v_2, v_3, v_4\}, \\
 A &= \{\{\{s, v_2\}, \{s, v_3\}, \{v_1, v_2\}, \{v_1, v_4\}\}, \{v_2, v_3, v_4\}\}.
 \end{aligned}$$

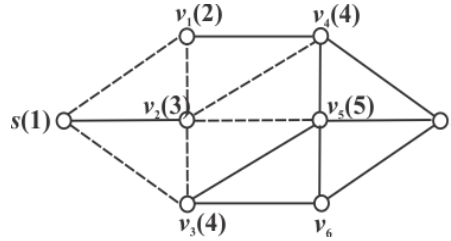
На Фигура 2.7 е прикажано пресек множеството $\{\{s, v_2\}, \{s, v_3\}, \{v_1, v_2\}, \{v_1, v_4\}\}$:



Фигура 2.7 Ненасочен двотерминален транспортен систем со пресек множество $C = \{\{s, v_2\}, \{s, v_3\}, \{v_1, v_2\}, \{v_1, v_4\}\}$.

$$\begin{aligned}
 Z[2] &= v_1, F = \{v_1, v_2\}, \\
 D &= \{\{s, v_1\}, \{s, v_3\}, \{v_2, v_1\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_2, v_5\}\} \text{ -- пресек множество} \\
 H &= \{v_3, v_4, v_5\}, \\
 A &= \{\{\{\{s, v_2\}, \{s, v_3\}, \{v_1, v_2\}, \{v_1, v_4\}\}, \{v_2, v_3, v_4\}\}, \{\{\{s, v_1\}, \{s, v_3\}, \{v_2, v_1\}, \\
 &\quad \{v_2, v_4\}, \{v_2, v_3\}, \{v_2, v_5\}\}, \{v_3, v_4, v_5\}\}\}.
 \end{aligned}$$

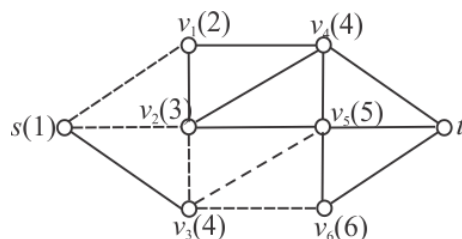
На Фигура 2.8 е прикажано пресек множеството $\{\{s, v_1\}, \{s, v_3\}, \{v_2, v_1\}, \{v_2, v_4\}, \{v_2, v_3\}, \{v_2, v_5\}\}$:



Фигура 2.8 Ненасочен двотерминален транспортен систем со пресек множество $C = \{\{s, v_1\}, \{s, v_3\}, \{v_2, v_1\}, \{v_2, v_4\}, \{v_2, v_3\}, \{v_2, v_5\}\}$.

$$\begin{aligned}
 Z[3] &= v_3, F = \{v_1, v_2, v_3\}, \\
 D &= \{\{s, v_1\}, \{s, v_2\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_6\}\}, \text{ -- пресек множество} \\
 H &= \{v_5, v_6\}, \\
 A &= \{\{\{\{s, v_2\}, \{s, v_3\}, \{v_1, v_2\}, \{v_1, v_4\}\}, \{v_2, v_3, v_4\}\}, \{\{\{s, v_1\}, \{s, v_3\}, \\
 &\quad \{v_2, v_1\}, \{v_2, v_4\}, \{v_2, v_3\}, \{v_2, v_5\}\}, \{v_3, v_4\}\}, \{\{\{s, v_1\}, \{s, v_3\}, \{v_2, v_3\}, \\
 &\quad \{v_3, v_5\}, \{v_3, v_6\}\}, \{v_5, v_6\}\}\}.
 \end{aligned}$$

На Фигура 2.9 е прикажано пресек множествата $\{\{s, v_1\}, \{s, v_2\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_6\}\}$:



Фигура 2.9 Ненасочен двотерминален транспортен систем со пресек множество $C = \{ \{s, v_1\}, \{s, v_2\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_6\} \}$

Алгоритмот продолжува на овој начин се додека сите пресек множества не се добијат. На крај од сите добиени пресек множества се бришат оние што не се минимални.

Со претходниот алгоритм се добиваат кандидати за минимални пресек множества, што не се минимални. Со цел да не добиваат вакви кандидати, анализираме на кој начин пресек множествата се добиени.

Пример 2.3 Да го разгледаме пресек множеството $\{ \{s, v_1\}, \{s, v_3\}, \{v_2, v_1\}, \{v_2, v_4\}, \{v_2, v_3\}, \{v_2, v_5\}, \{v_3, v_4, v_5\} \} \in A$. Ова пресек множество е прикажано на Фигура 2.8. Со користење на ова множество ќе добиеме нови кандидати за минимални пресек множества.

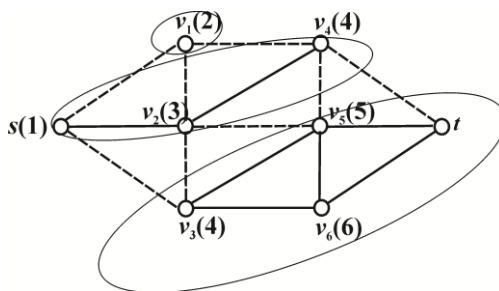
$$\begin{aligned} Z[2] &= v_4, F = \{v_1, v_2, v_3, v_4\} \\ D &= \{ \{s, v_1\}, \{s, v_3\}, \{v_2, v_1\}, \{v_2, v_3\}, \{v_2, v_5\}, \{v_1, v_4\}, \{v_4, v_5\}, \{v_4, t\} \}, \\ H &= \{v_5\}. \end{aligned}$$

Множеството $\{ \{s, v_1\}, \{s, v_3\}, \{v_2, v_1\}, \{v_2, v_3\}, \{v_2, v_5\}, \{v_1, v_4\}, \{v_4, v_5\}, \{v_4, t\}, \{v_5\} \}$ не е минимално пресек множество, бидејќи множеството $\{ \{s, v_3\}, \{v_3, v_2\}, \{v_2, v_5\}, \{v_5, v_4\}, \{v_4, t\} \}$ е помало пресек множество.

Од овие причини во Чекор 4 од Алгоритмот 2.2 треба да ги избришеме сите кандидати за минимални пресек множества што не се минимални. Ова може да биде направено со најмали $\ln(N)$ споредби помеѓу множествата, каде N е бројот на елементи на CutSet веднаш пред да влезе во Чекор 4. Со цел да не се појавуваат кандидати за минимални пресек множества што не се минимални, ние ќе направиме модификација на претходниот алгоритм.

Од Тврдењето 2.2 имаме дека кога добиеното пресек множество не е минимално, графот $G(V, E)$ е поделен во повеќе од две поврзани компоненти, како што е прикажано во Фигура 2.10.

Пресек множеството $\{ \{s, v_1\}, \{s, v_3\}, \{v_2, v_1\}, \{v_2, v_3\}, \{v_2, v_5\}, \{v_1, v_4\}, \{v_4, v_5\}, \{v_4, t\} \}$ не е минимално пресек множество и го дели графот $G(V, E)$ на три компоненти, како што е прикажано на Фигура 2.10:



Фигура 2.10 Ненасочен двотерминален транспортен систем со пресек множество
 $C = \{\{s, v_1\}, \{s, v_3\}, \{v_2, v_1\}, \{v_2, v_3\}, \{v_2, v_5\}, \{v_1, v_4\}, \{v_4, v_5\}, \{v_4, t\}\}.$

Идејата на новиот алгоритам е да се провери дали со бришењето на линковите од новиот кандидат за минимално пресек множество, добиениот подграф е поделен на две сврзани компоненти, такашто изворот е во едната од нив, а сливникот во другата. Значи според Тврдењето 2.1 треба да се провери дали графовите $(B_3, \{\{u, v\} \mid u, v \in B_3\})$ и $(V / B_3, \{\{u, v\} \mid u, v \in V / B_3\})$ се сврзани графови.

Твдрење 2.4 Во секоја итерација (повторување) во Чекор 3.1, графот што се добива $G(B_3, \{\{u, v\} \mid u, v \in B_3\})$ е сврзан граф.

Доказ: Иницијално имаме дека $B_3 = \{s\}$, од овде е јасно дека пред влегувањето во циклусот во Чекор 3.1, графот $G(B_3, \{\{u, v\} \mid u, v \in B_3\})$ е сврзан граф. Ние ќе покажеме дека тврдењето “ $G(B_3, \{\{u, v\} \mid u, v \in B_3\})$ е сврзан граф” е инваријанта на циклусот во Чекор 3.1.

Претпоставуваме дека пред влегување во циклусот во Чекор 3.1, $G(B_3, \{\{u, v\} \mid u, v \in B_3\})$ е сврзан граф. Ние треба да покажеме дека графот $G(B_3', \{\{u, v\} \mid u, v \in B_3'\})$ е сврзан граф. Нека $B_3' = B_3 \cup \{v_j\}$ и $B_2' = \{B_2 / \{v_j\} \cup \{v_i \mid \{v_i, v_j\} \notin B_1\}\}$. Од $v_j \in B_2$, следува дека v_j е поврзано со најмалку еден елемент од B_3 , значи графот $G(B_3 \cup \{v_j\}, \{\{u, v\} \mid u, v \in B_3 \cup \{v_j\}\})$ е сврзан граф. \square

Останува уште да провериме дека графот $(V/B_3, \{\{u, v\} \mid u, v \in V/B_3\})$ е сврзан граф. Поради тоа на почетокот ние ќе конструираме дрво со корен во сливникот. Во Чекор 3.3.1.1 од Алгоритам 2.2, кога бираме нов јазол v_j ќе проверуваме дали тој јазол е лист во дрвото.

- Кога јазолот v_j е лист, тогаш е јасно дека подграфот ги има потребните особини. Па, ние конструираме друго дрво со бришење на овој лист, и постапката ја продолжуваме од Чекор 3.3.1.2 до Чекор 3.3.1.4 од Алгоритамот 2.2.
- Кога јазолот v_j не е лист, го бришеме овој јазол од дрвото и проверуваме дали јазлите што се негови потомци се поврзани со друг јазол од остатокот на дрвото. Ако сите јазли се поврзани со друг јазол од остатокот на дрвото, тогаш добиеното множество е минимално пресек множество, па ние ќе конструираме друго дрво и ќе продолжиме со постапката од Чекор 3.3.1.2 до Чекор 3.3.1.4 од Алгоритамот 2.2. Ако најмалку еден јазол не е поврзан со остатокот од дрвото, тогаш множеството не е минимално пресек множество и земаме друг елемент од A .

Може да се забележи дека сите линкови на графот се значајни, т.е. графот како систем е кохерентен систем, па соодветното минимално пресек множество кое е подмножество од некое пресек множество кое е добиено со алгоритмот ќе се добие и на друг начин. Тоа значи дека ако графот е кохерентен, тогаш секогаш кога ќе добиеме кандидат за минимално пресек множество кое не е минимално пресек множество, тоа множество можеме да не го додаваме во A .

Сега ќе го дадеме псевдокодот на новиот подобрен алгоритам.

Алгоритам 2.3 – Модифициран алгоритам за пресметување на множество од минимални пресеци ($G(V, E)$)

Чекор 1. Се иницијализираат A и \mathcal{T} , $A = \{\{\{s, v_i\} \mid \{s, v_i\} \in E\}, \{v_i \mid \{s, v_i\} \in E\}, \mathcal{T}\}$, \mathcal{T} е дрво со корен во сливникот,

Чекор 2. Се иницијализира CutSet, $\text{CutSet} = \{\{s, v_i\} \mid \{s, v_i\} \in E\}$,

Чекор 3. Додека $A \neq \emptyset$, се зема елемент од множество A , $B = \{B_1, B_2, \mathcal{T}\}$ и се брише тој елемент од A ,

Чекор 3.1 За секој елемент $v_j \in B_2$, $v_j \neq t$, се проверува дали v_j е лист.

Ако v_j е лист тогаш

$$B_2 = \{B_2 / \{v_j\} \cup \{v_i \mid \{v_i, v_j\} \notin B_1\},$$

$$B_1 = \{B_1 / \{\{v_i, v_j\} \mid \{v_i, v_j\} \in B_1\} \cup \{\{v_i, v_j\} \mid \{v_i, v_j\} \notin B_1\},$$

$$\mathcal{T} = \mathcal{T} / \{v_j\},$$

$$\text{CutSet} = \text{CutSet} \cup \{B_1\},$$

во спротивно

ако сите јазли v_j' што се наследници на јазолот v_j се поврзани со друг јазол од остатокот од дрвото тогаш

$$B_2 = \{B_2 / \{v_j\} \cup \{v_i \mid \{v_i, v_j\} \notin B_1\},$$

$$B_1 = \{B_1 / \{\{v_i, v_j\} \mid \{v_i, v_j\} \in B_1\} \cup \{\{v_i, v_j\} \mid \{v_i, v_j\} \notin B_1\},$$

$$\text{CutSet} = \text{CutSet} \cup \{B_1\},$$

Новите линкови што го поврзуваат јазолот v_j' со остатокот од дрвото \mathcal{T} се додаваат во дрвото и се бришат старите линкови, во спротивно

ако најмалку еден јазол не е поврзан со остатокот од дрвото, тогаш оди на Чекор 3.1.

$$A = A \cup \{\{B_1, B_2, \mathcal{T}\}\}.$$

Чекор 4. Врати го CutSet.

Следниот пример ќе ја илустрира работата на Алгоритам 2.3

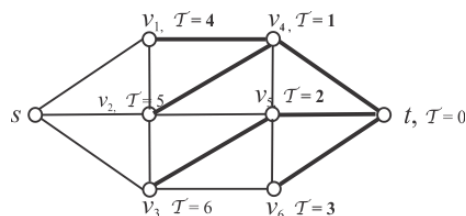
Пример 2.4

1. Кога јазолот е лист

$$A = \{\{\{s, v_1\}, \{s, v_2\}, \{s, v_3\}\}, \{v_1, v_2, v_3\}\},$$

$$B_3 = \{s\}, V / B_3 = \{v_1, v_2, v_3, v_4, v_5, v_6\}.$$

Фигурата 2.11 го прикажува соодветното дрво $\{\{t, \{v_4, v_5, v_6\}\}, \{v_4, \{v_1, v_2\}\}, \{v_5, \{v_3\}\}, \{v_6, \{\}\}, \{v_1, \{\}\}, \{v_2, \{\}\}, \{v_3, \{\}\}\}$.

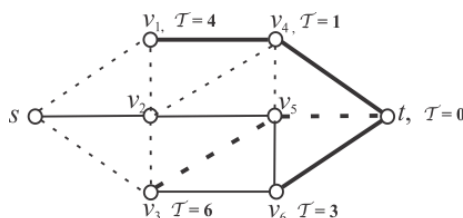


Фигура 2.11 Ненасочен двотерминален транспортен систем со дрво

$$\mathcal{T} = \{\{t, \{v_4, v_5, v_6\}\}, \{v_4, \{v_1, v_2\}\}, \{v_5, \{v_3\}\}, \{v_6, \{\}\}, \{v_1, \{\}\}, \{v_2, \{\}\}, \{v_3, \{\}\}\}.$$

Бидејќи јазолот v_6 е лист, тој може да биде избришан од дрвото. Јасно е дека остатокот од дрвото е сврзан.

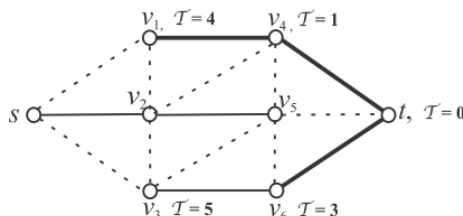
- Кога јазолот не е лист, а сите негови наследници се поврзани со друг јазол од остатокот на дрвото. Нека почетното пресек множество е $\{\{s, v_1\}, \{s, v_3\}, \{v_2, v_3\}, \{v_2, v_5\}, \{v_2, v_1\}, \{v_2, v_4\}\}, \{v_3, v_4, v_5\}$ и го разгледуваме јазолот v_5 . Соодветното дрво за ова множество е $\{t, \{v_4, v_5, v_6\}\}, \{v_4, \{v_1\}\}, \{v_5, \{v_3\}\}, \{v_6, \{\}\}, \{v_1, \{\}\}, \{v_3, \{\}\}$. По бришењето на јазолот v_5 ќе го добиеме дрвото $\{t, \{v_4, v_6\}\}, \{v_4, \{v_1\}\}, \{v_6, \{\}\}, \{v_1, \{\}\}$, Фигура 2.12:



Фигура 2.12 Ненасочен двотерминален транспортен систем со дрво

$$\mathcal{T} = \{t, \{v_4, v_6\}\}, \{v_4, \{v_1\}\}, \{v_6, \{\}\}, \{v_1, \{\}\}.$$

Јазолот v_3 е наследник на јазолот v_5 . Овој јазол е поврзан со јазолот v_6 . Новото дрво $\{t, \{v_4, v_6\}\}, \{v_4, \{v_1\}\}, \{v_6, \{v_3\}\}, \{v_1, \{\}\}, \{v_3, \{\}\}$ е прикажано во Фигура 2.13. Добиено минимално пресек множество е $\{\{s, v_1\}, \{s, v_3\}, \{v_2, v_3\}, \{v_2, v_1\}, \{v_2, v_4\}, \{v_5, v_6\}, \{v_5, v_4\}, \{v_5, t\}\}$.

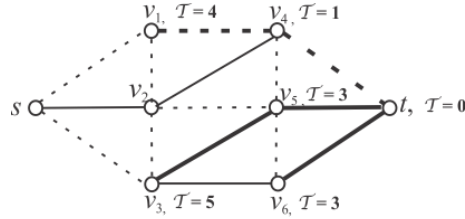


Фигура 2.13 Ненасочен двотерминален транспортен систем со дрво

$$\mathcal{T} = \{t, \{v_4, v_6\}\}, \{v_4, \{v_1\}\}, \{v_6, \{v_3\}\}, \{v_1, \{\}\}, \{v_3, \{\}\}.$$

- Кога јазолот не е лист и најмалку еден јазол што е негов наследник не е поврзан со остатокот од дрвото. Го разгледуваме множеството $\{\{s, v_1\}, \{s, v_3\}, \{v_2, v_3\}, \{v_2, v_5\}, \{v_2, v_1\}, \{v_2, v_4\}\}, \{v_3, v_4, v_5\}$ и го земаме јазолот v_4 . Соодветното дрво за ова множество е $\{t, \{v_4, v_5, v_6\}\}, \{v_4, \{v_1\}\}, \{v_5, \{v_3\}\}, \{v_6, \{\}\}, \{v_1, \{\}\}$.

$\{\}, \{v_3, \{\}\}\}$. По бришењето на јазолот v_4 , (линковите $\{v_1, v_4\}$, $\{v_4, t\}$) ќе го добијеме дрвото $\{t, \{v_5, v_6\}\}$, $\{v_5, \{v_3\}\}$, $\{v_6, \{\}\}$, $\{v_3, \{\}\}$, Фигура 2.14:



Фигура 2.14 Ненасочен двотерминален транспортен систем со дрво $\mathcal{T} = \{t, \{v_5, v_6\}\}$, $\{v_5, \{v_3\}\}$, $\{v_6, \{\}\}$, $\{v_3, \{\}\}\}$.

Јазолот v_1 е наследник на јазолот v_4 . Овој јазол не е поврзан со јазлите v_5 и v_6 , т.е. не е поврзан со јазол од остатокот од дрвото. Значи дека кандидатот за минимално пресек множество не е минимално пресек множество.

2.3.2 Алгоритам за определување на минимални пресек множества на ненасочени мрежи со користење на BFS

Во ова подглавје ќе го разгледаме алгоритмот предложен во [39]. Повторно разгледуваме ненасочен двотерминален транспортен систем $G(V, E)$, каде V е множество од јазли и E е множество од линкови. Според Тврдењето 2.2 од претходниот дел, следува дека со бришење на линковите од некое минимално пресек множество C , графот ќе биде поделен на точно две сврзани компоненти, такашто s ќе припаѓа во една од нив а t ќе припаѓа во другата. Ако го разгледаме графот што е прикажан на Фигура 2.5, можеме да видиме дека пресек множеството $C = \{\{s, v_3\}, \{v_3, v_2\}, \{v_2, v_5\}, \{v_5, v_4\}, \{v_4, t\}\}$ го дели графот на два графови (две сврзани компоненти), $G_1(\{s, v_1, v_2, v_4\}, \{\{s, v_1\}, \{s, v_2\}, \{v_1, v_4\}, \{v_2, v_4\}, \{v_1, v_4\}\})$ и $G_2(\{v_3, v_5, v_6, t\}, \{\{v_3, v_5\}, \{v_3, v_6\}, \{v_5, v_6\}, \{v_5, t\}, \{v_6, t\}\})$. Идејата на алгоритмот е да се конструираат сите овие сврзани компоненти.

Со цел да се отстрани појавувањето на ист сврзан граф повеќе од еднаш, ние дефинираме подредување на јазлите. За ова цел ќе го користиме пребарувањето по широчина (*breadth-first search* (BFS)) што започнува од изворот s и ја дефинираме функцијата:

$$\mathcal{F}: V \rightarrow \mathbb{N}, \mathcal{F}(v_i) = i \text{ ако } v_i \text{ е } i - \text{от јазол во BFS пребарувањето што започнува од } s. \quad (2.19)$$

Значи, ако имаме дека $\mathcal{F}(v_i) < \mathcal{F}(v_j)$, тогаш најкраткиот пат од s до v_j е подолг или еднаков со најкраткиот пат од s до v_i . Во секој чекор во “while (додека)” циклусот се користат елементите $\{B_1, B_2, B_3, \mathcal{T}\}$. Постои разлика помеѓу значењето на елементите на ова множество и истите множества што ги користевме во претходниот дел, па затоа повторно да го разгледаме нивното значење што ќе го користиме во новопредложениот алгоритам. B_1 е едно минимално пресек множество за дадениот граф. Соодветниот граф G_1 е $G_1(B_3, \{\{u, v\} \mid u, v \in B_3\})$ и B_2 е множеството од јазли v од V/B_3 што треба да се додадат во множеството B_3 во некој од наредните чекори. \mathcal{T} е дрво со корен во t и ги содржи сите

јазли што не се во B_3 . Во дрвото \mathcal{T} се чуваат множества од јазли, листата од наследници на секој јазол ја означуваме со DescadentList и родителската функција ја означуваме со π .

На почеток ги иницијализираме множествата B_1, B_2, B_3 . B_1 е минимално пресек множество кое ги содржи сите линкови со почетен јазол s , B_3 е иницијализирано на $\{s\}$ и B_2 е множество од сите јазли кои се крајни јазли на линковите чиј што почетен јазол е s . Додека дрвото \mathcal{T} е DFS дрво на графот $G'(V/B_3, E/B_1)$ со корен во сливникот t . Јасно е дека минималното пресек множество B_1 го дели графот на два графови $G_1(\{s\}, \emptyset)$ и $G_2(V/\{s\}, E/B_1)$.

Алгоритам 2.4 – Програма за пресметување на множество од минимални пресеци која ја повикува процедурата CONNECTION ($G(V, E)$)

Чекор 1. Се конструира функцијата \mathcal{F} со користење на BFS кое започнува во s ;

Чекор 2. Се иницијализира множеството $B_1 = \{\{s, v_i\} \mid \{s, v_i\} \in E\}$,

Чекор 3. Се иницијализира множеството $B_2 = \{v_i \mid \{s, v_i\} \in E\} / \{t\}$,

Чекор 4. Се иницијализира множеството $B_3 = \{s\}$,

Чекор 5. Се иницијализира дрвото $\mathcal{T} = \{\text{DescadentList}, \pi\}$ на графот $G'(V/B_3, E/B_1)$ со BFS,

Чекор 6. $A = \{\{B_1, B_2, B_3, \mathcal{T}\}\}$,

Чекор 7. Се иницијализира MinCutSet = $\{\{s, v_i\} \mid \{s, v_i\} \in E\}$,

Чекор 8. Додека $A \neq \emptyset$ извршувај

Чекор 8.1 Се зема елемент $B = \{B_1, B_2, B_3, \mathcal{T}\} \in A$ и се брише од A

Чекор 8.1.1 За секој елемент $v_j \in B_2$ извршувај

Чекор 8.1.1.1 Ако first(CONNECTION(E, B_1, v_j, \mathcal{T})) = True тогаш

Чекор 8.1.1.1.1 $B_3 = B_3 \cup \{v_j\}$

Чекор 8.1.1.1.2 $B_2 = \{x \mid x \in B_2 \wedge \mathcal{F}(v_j) < \mathcal{F}(x)\} \cup \{v_i \mid \{v_i, v_j\} \notin B_1 \wedge (\mathcal{F}(v_j) < \mathcal{F}(x) \vee \neg (\exists \{x, v_i\}, \{x, v_i\} \in B_1))\}$,

Чекор 8.1.1.1.3 $B_1 = (B_1 / \{\{v_i, v_j\} \mid \{v_i, v_j\} \in B_1\}) \cup \{\{v_i, v_j\} \mid \{v_i, v_j\} \notin B_1\}$,

Чекор 8.1.1.1.4 MinCutSet = MinCutSet $\cup \{B_1\}$,

Чекор 8.1.1.1.5 $A = A \cup \{B_1, B_2, B_3, \text{Last(CONNECTION)}\}$

Чекор 8.1.1.2 Инаку ако MIN(CONNECTION[[2]]) > b

Чекор 8.1.1.2.1 $B_4 = \text{CONNECTION}[[2]]$,

Чекор 8.1.1.2.2 $B_3 = B_3 \cup \{v_j\} \cup B_4$,

Чекор 8.1.1.2.3 $B_2 = \{x \mid x \in B_2 \wedge \mathcal{F}(v_j) < \mathcal{F}(x)\} \cup \{v_i \mid \{v_i, v_j\} \notin B_1 \wedge v_i \notin B_4 \wedge (\mathcal{F}(v_j) < \mathcal{F}(x) \vee \neg (\exists \{x, v_i\}, \{x, v_i\} \in B_1))\}$,

Чекор 8.1.1.2.4 $B_1 = (B_1 / \{\{v_i, v_j\} \mid \{v_i, v_j\} \in B_1\}) \cup \{\{v_i, v_j\} \mid \{v_i, v_j\} \notin B_1 \wedge v_i \notin B_4\}$,

Чекор 8.1.1.2.5 MinCutSet = MinCutSet $\cup \{B_1\}$,

Чекор 8.1.1.2.6 $A = A \cup \{B_1, B_2, B_3, \text{Last(CONNECTION)}\}$,

Чекор 9 Врати го MinCutSet.

Во секое повторување на “додека (while)” циклусот во Чекор 8, земаме елемент $\{B_1, B_2, B_3, T\}$ од A . Пресек множеството B_1 го дели графот G во две поврзани компоненти $G_1(B_2, E_2)$ и $G_2(V/B_2, E/(E_2 \cup B_1))$. Тогаш во секоја итерација во “за (for)” циклусот во

(BFS) што започнува од сливникот t . Ако остатокот од G_2 е сврзан, во чекорите Чекор 8.1.1.1.1 - Чекор 8.1.1.1.5 ќе додадеме друг член во A и друго минимално пресек множество во MinCutSet.

Ако остатокот од G_2 не е сврзан, правиме дополнителни проверки. Бидејќи имаме сврзан граф, постои поврзување со некој јазол од $V \setminus V_{\mathcal{T}}$ од каде следува дека V и $V \setminus V_{\mathcal{T}}$ го делат графот во две сврзани компоненти. Поради тоа што ние не сакаме да се повторуваат исти комбинации од јазли во B_3 , ќе ја користиме стратегијата за додавање на јазли кои имаат поголема вредност за \mathcal{F} . Но не е секогаш можно да се додаваат јазли во растечки редослед, па затоа кога ќе добиеме множество од јазли што не се поврзани со t , тогаш во B_3 ќе ги додадеме оние јазли што имаат вредност за \mathcal{F} поголема од $\mathcal{F}(v_j)$ (Чекор 8.1.1.2.1 - Чекор 8.1.1.2.6).

Во Алгоритмот 2.4 се повикува процедурата CONNECTION. Сега ќе објасниме како работи оваа процедура. Влез на процедурата CONNECTION се графот G , дрвото \mathcal{T} и еден јазол v_j од \mathcal{T} . Со процедурата се проверува дали сите јазли во дрвото \mathcal{T} се поврзани. Ако после бришењето на јазолот v_j , постојат јазли во \mathcal{T} што не се поврзани со сливникот t и сите тие имаат помала вредност за \mathcal{F} од $\mathcal{F}(v_j)$, тогаш процедурата CONNECTION ќе врати излез FAULT. Во спротивно, процедурата CONNECTION ќе врати излез TRUE заедно со новото дрво со корен во t и листа од неповрзани јазли. На почетокот, во процедурата CONNECTION (Чекор 1) се проверува дали јазолот v_j е лист во дрвото \mathcal{T} . Ако v_j е лист, тогаш дрвото што се добива од \mathcal{T} со бришење на v_j ги поврзува останатите јазли од \mathcal{T} . Кога v_j не е лист, мораме да направиме дополнителни проверки. Од Чекор 2.4.1 до Чекор 2.4.4, секој наследник на v_j се обојува во црвено и се брише од $V_{\mathcal{T}}$. Тогаш за секој јазол што е обоен во црвено се проверува дали постои линк помеѓу него и некој друг јазол од $V_{\mathcal{T}}$ (јазлите за кои веќе е познато дека се поврзани со сливникот t). Ако постои таков линк, соодветниот јазол се обојува во сиво и се додава во дрвото (v_l е родителски јазол на јазолот v_k и v_k внесува во листата DescendentList на v_l , Чекор 2.5.1.2 и Чекор 2.5.1.3. Според ова, ако не постојат јазли што се обоени во сиво, јасно е дека јазлите во $V_{\mathcal{T}}$ не се поврзани, во спротивно сè уште не сме сигурни дали јазлите се поврзани и мораме да направиме дополнителни проверки. За таа цел ќе го користиме пребарувањето по широчина, кое започнува од јазлите што се обоени во сиво и се додаваат во дрвото сите јазли обоени во сиво до кои се стигнува се пребарувањето. Ако сè уште постојат јазли што се обоени во црвено, тогаш постојат јазли што не се во $V_{\mathcal{T}}$, што значи дека сите јазли не се поврзани и променливата *isconnect* ќе добие вредност False; во спротивно јазлите се поврзани и променливата *isconnect* ќе добие вредност True.

Алгоритам 2.5 – Процедура CONNECTION ($G(V, E)$, $\mathcal{T} = (V_{\mathcal{T}}, \text{DescendentList}, \pi)$, v_j)

Чекор 1. Ако $\text{DescendentList}(b) = \emptyset$ тогаш //ако v_j е лист

Чекор 1.1 *isconnect* = True,

Чекор 1.2 Избриши го v_j од $\text{DescendentList}(\pi(v_j))$,

Чекор 1.3 $\pi(v_j) = \text{NIL}$.

Чекор 2. Инаку

Чекор 2.1 $\text{Gray} = \emptyset$,

Чекор 2.2 $\mathcal{M} = \text{Red} = \text{DescendentList}(v_j)$,

Чекор 2.3 $V_{\mathcal{T}} = V_{\mathcal{T}} / \mathcal{M}$.

Чекор 2.4 Додека $\mathcal{M} \neq \emptyset$ извршувај

Чекор 2.4.1 $v_k = \text{First}(\mathcal{M})$,

Чекор 2.4.2 $\mathcal{M} = (\mathcal{M} / \{v_k\}) \cup \text{DescendentList}(v_k)$,

Чекор 2.4.3 $\text{Red} = \text{Red} \cup \text{DescendentList}(v_k)$,

Чекор 2.4.4 $V_{\mathcal{T}} = V_{\mathcal{T}} / \text{DescendentList}(v_k)$.

Чекор 2.5 За сите $v_k \in \text{Red}$ извршувај

Чекор 2.5.1 Ако постои v_l такашто $\{v_k, v_l\} \in E$ и $v_l \in V_{\mathcal{T}}$ тогаш

Чекор 2.5.1.1 $\text{Gray} = \text{Gray} \cup \{v_k\}$,

Чекор 2.5.1.2 $\pi(v_k) = v_l$,

Чекор 2.5.1.3 $\text{DescendentList}(v_l) = \text{DescendentList}(v_l) \cup \{v_k\}$.

Чекор 2.6 Додека $\text{Gray} \neq \emptyset$ извршувај

Чекор 2.6.1 $v_m = \text{First}(\text{Gray})$,

Чекор 2.6.2 $\text{Gray} = \text{Gray} / \{v_m\}$,

Чекор 2.6.3 $V_{\mathcal{T}} = V_{\mathcal{T}} \cup \{v_m\}$,

Чекор 2.6.4 $\text{DescendentList}(v_m) = \{u \mid \{u, v_m\} \in V \text{ и } u \in \text{Red}\}$,

Чекор 2.6.5 $\text{Gray} = \text{Gray} \cup \text{DescendentList}(v_m)$,

Чекор 2.6.6 $\text{Red} = \text{Red} / \text{DescendentList}(v_m)$,

Чекор 2.6.7 За сите $u \in \text{DescendentList}(v_m)$ изврши $\pi(u) = v_m$,

Чекор 2.7 Ако $\text{Red} \neq \emptyset$ тогаш $\text{isconnect} = \text{False}$ инаку $\text{isconnect} = \text{True}$

Чекор 3. Ако $\text{isconnect} = \text{False}$ тогаш $\text{output} = \{\text{False}, \text{Red}, \mathcal{T} = (V_{\mathcal{T}}, \text{DescendentList}, \pi)\}$ инаку
 $\text{output} = \{\text{True}, \mathcal{T} = (V_{\mathcal{T}}, \text{DescendentList}, \pi)\}$

Чекор 4. Врати го output .

Пример 2.5 Да разгледаме како алгоритмот работи на транспортниот систем даден на Фигура 2.15 со извор 1 и сливник 7. Дрвото со корен во сливникот 7 е прикажано на Фигура 2.15 b. На почеток $A = \{\{B_1 = \{1, 2\}, \{1, 3\}\}, B_2 = \{2, 3\}, B_3 = \{1\}, \mathcal{T} = \{\{7, \{2, 3\}\}, \{2, \{5, 4\}\}, \{3, \{6\}\}, \{4, \emptyset\}, \{5, \emptyset\}, \{6, \emptyset\}\}\}$.

Во првиот чекор, кога процедурата CONNECTION е повикана за првиот елемент 2, е прикажано како алгоритмот работи со незначајни (ирелевантни) линкови. Јазолот 2 не е лист, па јазлите 4 и 5 се ставаат во множеството Red и се бришат од \mathcal{T} . Јазолот 4 е поврзан со јазолот 6, па јазолот 4 се обојува во сиво и е поврзан со дрвото преку јазолот 6. Јазолот 5 не е поврзан со ниту еден јазол од дрвото \mathcal{T} , и по излегување од процедурата CONNECTION, $\text{Red} = \{5\}$. Новото дрво е $\{\{7, \{3\}\}, \{3, \{6\}\}, \{6, \{4\}\}, \{4, \emptyset\}\}$. Бидејќи $\min\{5\} = 5 > 2$, е добиено минималното пресек множество $\{\{1, 3\}, \{2, 4\}, \{2, 7\}\}$. Елементот

$$\{\{\{1, 3\}, \{2, 4\}, \{2, 7\}\}, \{3, 4\}, \{1, 2, 5\}, \{\{7, \{3\}\}, \{3, \{6\}\}, \{6, \{4\}\}, \{4, \emptyset\}\}\} \quad (2.20)$$

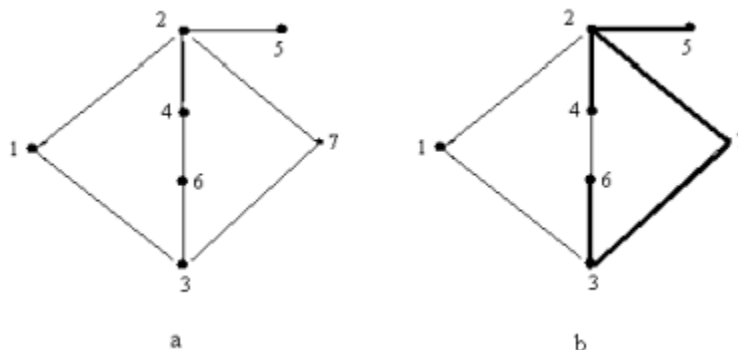
се додава во множеството A.

Во наредниот чекор процедурата CONNECTION се повикува за јазолот 3. Бидејќи јазолот 3 не е лист, јазолот 6 се внесува во множеството Red и се брише од \mathcal{T} . Но јазолот 6 е поврзан со јазолот 4 и Red станува \emptyset .

Елементот

$$\{\{1, 2\}, \{3, 6\}, \{3, 7\}\}, \{6\}, \{1, 3\}, \{\{7, \{2\}\}, \{2, \{4, 5\}\}, \{4, \{6\}\}, \{6, \emptyset\}, \{5, \emptyset\}\} \quad (2.21)$$

се додава во множеството A .



Фигура 2.15 Двотерминален транспортен систем со извор 1 и сливник 7.

Во следниот чекор, ќе се прикаже еден од карактеристичните случаи. Се зема елементот даден со (2.20) и процедурата CONNECTION се повикува за јазолот 3. Јазлите 6 и 4 се додаваат во множеството Red . Кога ќе се избрише 3, јазлите 6 и 4 не се поврзани со сливникот. Но поради тоа што $\min\{4, 6\} = 4 > 3$, јазлите 4 и 6 се додаваат во множеството B_3 . Бидејќи како минимално пресек множество се добива $\{\{2, 7\}, \{3, 7\}\}$ а $B_2 = \emptyset$, за овој елемент не се добива друго пресек множество.

Следниот случај е карактеристичен случај, кога се додава лист во B_3 . Поради тоа, процедурата CONNECTION се повикува за јазолот 4, кој што е лист во (2.20), и елементот

$$\{\{1, 3\}, \{6, 4\}, \{2, 7\}\}, \{6\}, \{1, 2, 4, 5\}, \{\{7, \{3\}\}, \{3, \{6\}\}, \{6, \emptyset\}\} \quad (2.22)$$

се додава во множеството A .

Откако завршивме со елементот (2.20) преминуваме на елементот (2.21). Бидејќи во B_2 елемент е само јазолот 6 кој е лист, се добива пресек множеството $\{\{1, 2\}, \{3, 7\}, \{6, 4\}\}$. Јазолот 4 е поврзан со јазолот 6 кој што е јазол во множество од линкови $\{\{1, 2\}, \{3, 6\}, \{2, 7\}\}$. Го додаваме јазолот 4 во множеството B_2 и елементот

$$\{\{1, 2\}, \{4, 6\}, \{3, 7\}\}, \{4\}, \{1, 3, 6\}, \{\{7, \{2\}\}, \{2, \{4, 5\}\}, \{4, \emptyset\}, \{5, \emptyset\}\}. \quad (2.23)$$

Сега го разгледуваме (2.22), и го додаваме јазолот 6 во B_3 . B_2 станува празно, бидејќи јазолот 6 е поврзан со јазолот 4 кој е во B_3 и јазолот 3 кој е јазол во B_1 . Од овде се добива минималното пресек множество $\{\{1, 3\}, \{6, 3\}, \{2, 7\}\}$. Слично од (2.23) се добива пресек множеството $\{\{1, 2\}, \{4, 2\}, \{3, 7\}\}$ и со тоа алгоритмот завршува.

Сега ќе покажеме дека предложениот алгоритам за наоѓање на минимални пресек множества на ненасочена мрежа работи коректно.

Тврдење 2.5 Нека MINIMAL_CUT_SET е ограничено на графот $G(V, E)$. После секоја итерација во “додека” циклусот во Чекор 8, графот $G_1(B_3, \{u, v\} \mid u, v \in B_3)$ е сврзан граф така што $s \in B_3$.

Доказ: Јасно е дека $G_1(\{s\}, \emptyset)$ е сврзан граф. Во секоја итерација на циклусот земаме елемент од множеството $A = \{B_1, B_2, B_3, T\}$ и се додава нов елемент во множеството B_3 . Се претпоставува дека за сите цели броеви помали или еднакви на k ако $|B_3| < k$ тогаш графот $G_1(B_3, \{u, v\} \mid u, v \in B_3)$ е сврзан граф. Нека B_3 е множество кое се добива со додавање на ново множество од јазли $\text{Red} \cup \{v\}$ на множеството B_3 , за кое важи $|B_3| < k$. Јасно е дека подграфот од G што ги содржи овие јазли е поврзан. Уште повеќе, бидејќи G е сврзан граф, овој граф е поврзан со некој јазол $u \in B_3$. Значи B_3 е исто така сврзано множество и $s \in B_3 \subset B_3$. \square

Тврдење 2.6 Нека процедурата CONNECTION е повикана од сврзаниот граф G , дрвото \mathcal{T} од јазли од G , во корен во сливникот t и јазолот $v_j \neq t$ од \mathcal{T} . Нека $G_2(\{u \mid u \text{ е во } \mathcal{T}\} / \{v_j\}, \{\{u, v\} \mid u, v \in \{u \mid u \text{ е во } \mathcal{T}\} / \{v_j\}\})$, тогаш CONNECTION добива вредност False кога G_2 не е сврзан граф, а добива вредност True заедно со дрвото што ги поврзува јазлите од G_2 , кога G_2 е сврзан граф.

Доказ: Прво се претпоставува дека G_2 не е сврзан граф. Тогаш постои јазол u за кој не постои линк од t до u во G_2 . Ние тврдиме дека сите патишта во $G_2(\{u \mid u \text{ е во } T\}, \{\{u, v\} \mid u, v \in \{u \mid u \text{ е во } \mathcal{T}\}\})$ поминуваат низ v_j (ако тоа не е точно, тогаш ќе постои пат во графот G_2 од t до u). Значи јазолот u мора да биде наследник на јазолот v_j во \mathcal{T} и u се обојува црвено со Чекор 2.4.3 и се брише од $V_{\mathcal{T}}$ со Чекор 2.4.4. Бидејќи u не е поврзан со ниту еден јазол од $V_{\mathcal{T}}$, u останува обоено црвено и после Чекор 2.6.7. Сега, можеме да тврдиме дека не постои пат од некој јазол v обоен во сиво до јазолот u . Во спротивно, доколку постои пат од u до v , $t \rightarrow v_i \rightarrow u \rightarrow v$, $\{v_j, u\} \in E$, $v_j \in V_{\mathcal{T}}$ тогаш јазолот v_j не е на овој пат што е контрадикција со претпоставката. Па, во Чекор 2.7 множеството $\text{Red} \neq \emptyset$ и процедурата CONNECTION ќе врати одговор False .

Сега, да претпоставиме дека G_2 е сврзан граф. Ние треба да докажеме дека по завршувањето на процедурата CONNECTION , $\text{Red} = \emptyset$. Бидејќи CONNECTION се повикува за јазол v_j што не е лист, по Чекор 2.5, $\text{Red} \neq \emptyset$. Нека $u \in \text{Red}$. Бидејќи G_2 е сврзан граф, постои пат од t до u во G_2 што не поминува низ v_j . Да претпоставиме дека v_i е првиот јазол што лежи на патот од t до u и v_i е наследник на v_j во \mathcal{T} . Јазолот v_i ќе го додадеме во новото дрво во Чекор 2.5.1.1 и во Чекор 2.5.1.3, јазолот u ќе го додадеме во новото дрво во чекорите на Чекор 2.6, бидејќи овој дел од алгоритмот е дел од BFS пребарувањето. \square

Теорема 2.3 Со програмата MINIMAL_CUT_SET се добиваат сите минимални пресек множества за даден граф $G(V, E)$.

Доказ: Од Тврдење 2.5 и Тврдење 2.6 следува дека секој елемент што го ставаме во MinCutSet го дели графот G во две сврзани компоненти, такашто s е во едната од нив а t во

другата. Значи секоја елемент во MinCutSet е минимално пресек множество. Останува уште да се докаже дека сите минимални пресек множества се содржат во MinCutSet. Тоа ќе го докажеме со индукција во однос на бројот на елементи на B_3 .

Јасно е дека важи за $k = 1$, бидејќи ако B_3 се состои само од еден елемент, тогаш $B_3 = \{s\}$. Да претпоставиме дека сите минимални пресек множества што го делат V на B_3 и V/B_3 , за кои $|B_3| \leq k$, се содржат во MinCutSet. Нека C го дели графот G во два сврзани графови $G_1(B_3, \{\{x, y\} \mid x, y \in B_3\})$ и $G_1(V/B_3, \{\{x, y\} \mid x, y \notin B_3\})$ и нека $v_j \in B_3$ таков што $\mathcal{F}(v_j) = \max\{\mathcal{F}(x) \mid x \in B_3 \text{ и постои линк } \{x, y\} \in E \text{ и } y \notin B_3\}$. Се дефинира $\mathcal{S}_0 = \{v_j\}$, $\mathcal{S}_k = \{x \in B_3 \mid \mathcal{F}(x) \geq \mathcal{F}(v_j) \text{ и } \{x, y\} \in E \text{ и } y \in \mathcal{S}_{k-1}\}$. Јасно е дека постојат конечно многу вакви множества, па нека $\mathcal{S} = \bigcup_k \mathcal{S}_k$ и нека множеството \mathcal{S}_1 е множество од сите соседни јазли на јазолот v_j во G_1 . Ќе разгледаме два случаја:

I случај: Со бришење на јазолот од \mathcal{S} , $G_1'(B_3/\mathcal{S}, \{\{x, y\} \mid x, y \in B_3/\mathcal{S}\})$ е сврзан. Овој случај ќе го поделиме на два под случаја, кога сите соседи на v_j во G_1 имаат поголема вредност на \mathcal{F} од $\mathcal{F}(v_j)$ и кога постои сосед x на v_j во G_1 за кој $\mathcal{F}(x) < \mathcal{F}(v_j)$. Да разгледаме прво како ќе го добиеме C кога сите соседи на v_j во G_1 имаат поголема вредност на \mathcal{F} од $\mathcal{F}(v_j)$ т.е. множеството од соседи на v_j во G_1 е \mathcal{S}_1 . Нека $v_i \in \mathcal{S}_1$ е јазол со помала вредност на \mathcal{F} и тој е поврзан со некој јазол од $B_3/\mathcal{S} \cup \{v_j\}$. Да го земеме најмалиот подграф на G_1 во којшто не се содржат јазолот v_j и сите негови соседи во $G_1/\{v_i\}$. Овој граф и остатокот од графот G (кој е сврзан бидејќи сите јазли во него се поврзани со v_j и v_j е поврзан со остатокот од графот) го делат G во две сврзани компоненти и сите линкови што не се во нив формираат минимално пресек множество. Според индуктивната претпоставка ова минимално пресек множество е добиено со алгоритмот. Уште повеќе, B_2 го содржи v_j (v_j се додава во вториот дел на формулата за B_2). Сега, ако $\mathcal{S} = \{v_j\}$, C се добива во делот од Чекор 2.4.4 до Чекор 2.5.1.1, во спротивно во делот од Чекор 2.6.1 до Чекор 2.6.5.

Во спротивно графот G_1' и остатокот од G го делат G во две сврзани компоненти и v_j се додава во B_2 во првиот дел на формулата за B_2). Значи повторно, ако $\mathcal{S} = \{v_j\}$, C се добива во делот од Чекор 2.4.4 до Чекор 2.5.1.1, во спротивно во делот од Чекор 2.6.1 до Чекор 2.6.5.

II случај: Со бришење на јазолот од \mathcal{S} , $G_1'(B_3/\mathcal{S}, \{\{x, y\} \mid x, y \in B_3/\mathcal{S}\})$ не е сврзан. Нека $\tilde{G}'(\tilde{V}', \tilde{E}')$ е сврзана компонента од G_1' во која припаѓа изворот s , и \tilde{G}' е графот $\tilde{G}(\tilde{V}' \cup \mathcal{S}, \{\{x, y\} \mid x, y \in \tilde{V}' \cup \mathcal{S}\})$. Значи за јазлите во \tilde{G}' кои имаат помала вредност за \mathcal{F} отколку $\mathcal{F}(v_j)$, пократкиот пат во G е пократок од пократкиот пат во \tilde{G}' . Овој пат мора да поминува низ јазли што не се во \tilde{G}' , што значи дека овие јазли се поврзани со некои други јазли од G_2 , т.е. \tilde{G} и $\tilde{G}_1(V/\tilde{V}' \cup \mathcal{S}, \{\{x, y\} \mid x, y \in V/(\tilde{V}' \cup \mathcal{S})\})$ го делат G во две сврзани компоненти и од индуктивната претпоставка соодветното минимално пресек множество е

добиемо. Јазлите со помала вредност за \mathcal{F} се додаваат во B_2 во вториот дел на формулата. Да го разгледаме графот \tilde{G}_1 . Нека v'_j е јазол во \tilde{G}_1 така што $\mathcal{F}(v'_j) = \max\{\mathcal{F}(x) \mid x \in B_3, x \in B_3 \cap (V / (\tilde{V}' \cup \mathcal{S}))\}$ и постои линк $\{x, y\} \in E$ и $y \notin B_3$. Се дефинира $\mathcal{S}'_0 = \{v'_j\}$, $\mathcal{S}'_k = \{x \in B_3 \cap (V / (\tilde{V}' \cup \mathcal{S})), \mid \mathcal{F}(x) \geq \mathcal{F}(v'_j) \text{ и } (x, y) \in E \text{ и } y \in \mathcal{S}'_{k-1}\}$ и $\mathcal{S}' = \bigcup_k \mathcal{S}'_k$.

Повторно имаме случај. Првиот случај е кога со бришење на јазлите од \mathcal{S}' остатокот на графот е сврзан, а вториот кога со бришење на јазлите од \mathcal{S}' остатокот на графот не е сврзан. Овие два случаја се разгледуваат на ист начин како и I случај и II случај. Оваа постапка се повторува се додека се добиват сврзани графови. Бидејќи G има конечен број јазли, процедурата ќе заврши. \square

Да ја разгледаме сложеноста на предложениот алгоритам. Секое минимално пресек множество се добива само еднаш, но во множеството од јазли B_3 се добиваат некои комбинации што не водат до минимално пресек множество. Добро е што секое множество B_3 е добиено најмногу еднаш. Најлош случај е кога сите подмножества на $V/\{t\}$ се добиваат од множеството B_3 . Ова се добива само за комплетни графови, но во овој случај секој кандидат за минимално пресек множество е всушност минимално пресек множество. Од овде ние имаме дека за граф со $|V|$ јазли процедурата CONNECTION се повикува најмногу $2^{|V|-1}$ пати.

Алгоритамот предложен во [1] има иста сложеност за наоѓање на минимални пресек множества, но кандидатите за минимални пресек множества што не се минимални се отфрлаат со меѓусебна споредба со сите други кандидати. Со оваа споредба се определува дали постои множество во листата од кандидати за минимални пресек множества кое е подмножество од дадено множество од линкови, и тоа треба да биде отстрането од листата. Со цел да се отстранат сите кандидати што не се минимални пресек множества од листа од N кандидати, потребно е да се направат $O(N^2)$ споредби помеѓу множествата.

Во предложениот алгоритам, ние имаме процедура за определување дали дадено пресек множество е минимално. Во оваа процедура ние правиме пребарување по широчина (BFS) во транспортниот систем. Во поголем дел од случаите постојат многу малку јазли на кои може да се прави BFS. Всушност, во густе графови, процедурата CONNECTION се повикува многу често. Меѓутоа во најголем број случаи, јазлите од кои се повикува процедурата се листови или имаат мал број наследници, со што сложеноста на процедурата е константна. На пример, за комплетен граф, процедурата CONNECTION се повикува само за листови.

Глава 3

3. Надежност на повеќе - состојбени транспортни системи и проточни транспортни системи

3.1 Надежност на повеќе - состојбени двотерминални транспортни системи

Во претходните поглавја дадовме краток осврт на методите за пресметување на надежност на бинарните транспортни системи. Надежноста на двотерминалните бинарни транспортни системи е доста проучуван проблем во теоријата на надежноста. Како што и претходно кажавме, компонентите на бинарниот систем, како и самиот систем, може да се најдат во една од две состојби, т.е. да бидат во работна состојба или да бидат расипани. Меѓутоа, токму оваа претпоставка дека системот и неговите компоненти може да бидат во една од две можни состојби доведува до грешка во претставувањето на вистинските модели на некои системи [46]. Системите како, системи за дистрибуција на вода, телекомуникациски системи, системи за снабдување со гас и нафта, системи за генерирање на енергија од водата и многу други, како и нивните составни компоненти може да се најдат во една од повеќе можни состојби. Затоа претпоставката дека ваквите системи може да се најдат во една од две можни состојби може да доведе до погрешно моделирање на системот, и овие модели во целост нема да го карактеризираат вистинското однесување на системот. Значи, да заклучиме, многу подобро ќе се моделираат овие системи доколку се претпостави дека самиот систем и неговите составни компоненти може да се најдат во една од повеќе состојби. Значи системот или некои негови компоненти може да не работат со полн капацитет, но сепак системот да функционира успешно, т.е. да биде во работна состојба.

Транспортните системи чии компоненти може да се најдат во една од повеќе состојби се нарекуваат **повеќе – состојбени двотерминални транспортни системи**. За повеќе – состојбените транспортни системи, проблемот за пресметување на надежноста на системите ќе биде анализиран под претпоставка дека системот и неговите компоненти

може да се најдат во една од повеќе можни состојби. Веднаш да нагласиме дека пресметувањето на надежноста на повеќе - состојбените транспортни системи е доста покомплексен проблем отколку пресметувањето на надежноста на бинарните двотерминални системи.

Да го разгледаме сега повеќе – сосостојбениот транспортен систем $G(V, E)$, каде V е множеството од јазли на системот, а E е множеството од линкови. Нека со s го означиме почетниот јазол, т.е. изворот, а со t крајниот јазол т.е. сливникот. Исто така, да претпоставуваме дека n е бројот на линкови на повеќе-состојбениот транспортен систем.

Повеќе-состојбен линк се дефинира како линк од транспортниот транспортниот систем кој има множество од состојби $\{0, s_1, s_2, \dots, M_i\}$, каде $0 < s_1 < s_2 \dots < M_i$, $1 \leq i \leq n$. Состојбата 0 соодветствува на случајот кога нема поток низ линкот. Состојбата M_i соодветствува на состојба со која линкот работи со полн капацитет. Останатите состојби се сите состојби во кои може да се најде линкот помеѓу состојбата кога е целосно расипан т.е. состојбата 0 и состојбата со која работи со полн капацитет. Векторот $\vec{M} = (M_1, M_2, \dots, M_n)$ е вектор од максимални состојби на линковите на транспортниот систем.

Нека x_i е состојбата во која се наоѓа линкот e_i , $1 \leq i \leq n$. Тогаш векторот $\vec{x} = (x_1, x_2, \dots, x_n)$ е вектор со кој се опишани сите состојби на системот и се нарекува **вектор на состојби**. За секој повеќе – состојбен линк се дефинира **вектор на капацитети** што претставува производ од полниот капацитет на линкот и векторот на состојби во кои може да се најде линкот. Нека векторот на состојби на i – от линк е $(0, 0.2, 0.4, 0.6, 0.8, 1)$ и ако кога линкот работи со полн капацитет низ него може да се пренесе поток од 5 единици, тогаш векторот на капацитети $S_i = (1, 2, 3, 4, 5)$. За транспортниот систем се дефинира множество од капацитет на системот S , што всушност претставува множество од сите можни капацитети од изворот s до сливникот t . Множеството од сите вектори на состојби се нарекува состојба на системот $S_1 \times S_2 \times \dots \times S_n$. Функцијата $\varphi: S_1 \times S_2 \times \dots \times S_n \rightarrow S$ која го пресликува векторот на состојби во состојба на системот се нарекува **повеќе – состојбена структурна** функција. Всушност $\varphi(\vec{x})$ е возможниот капацитет од изворот до сливникот, ако системот е во состојба \vec{x} .

Да ја дефинираме сега **повеќе – состојбената двотерминална надежност за ниво d** ($M2TR_d$). **Повеќе – состојбената двотерминална надежност за ниво d** ($M2TR_d$) е веројатноста дека поток поголем или еднаков на d , може да биде успешно пренесен од изворот до сливникот.

$$M2TR_d = P(\varphi(\vec{x}) \geq d) \quad (3.1)$$

Дефиниции за пат вектор, пресек вектор, минимален пат вектор и минимален пресек вектор се дадени во поглавјето 1.2.

За да може да се пресмета надежноста на системот треба да биде познато и множеството од веројатностите на нивоата на компонентите. Да го означиме ова множество со \mathcal{VP} . Ако \vec{p}_i е векторот од веројатности на нивоата на i – от линк, тогаш $p_{id} = P(x_i=d)$.

3.1.1 Пресметување на надежност на повеќе – состојбени транспортни системи

За пресметување на надежноста на повеќе – состојбените транспортни системи како и кај бинарните транспортни системи постојат повеќе методи за пресметување на надежноста. Како и кај бинарните транспортни системи и овде како наједноставен но најнеефикасен метод е методот кој се базира на набројување на сите можни состојби во кои може да се најдат компонентите од системот. Овој метод е неефикасен и кај релативно мали транспортни системи. На пример, ако имаме транспортен систем кој се состои од 8 линкови и секој од тие линкови може да се најде во една од четири можни состојби, тогаш вкупниот број на можни состојби во кои може да се најде системот е $4^8 = 65536$. Значи и за релативно мали транспортни системи, практично е невозможно да се набројат сите можни состојби. Постојат методи со кои се воведуваат некои услови, такашто се набројуваат само некои избрани состојби со кои се оценува надежноста, но и овие методи се покажале како доста неефикасни. Како најефикасен начин за пресметување на точната вредност на надежноста се покажал методот со користење на формулата за вклучување и исклучување. За да се користи оваа формула потребно е да се најдат сите минимални пат или минимални пресек вектори за дадено ниво. Затоа, целта на истражувачите е да пронајдат колку што е можно побрзи и поефикасни алгоритми за определување на минимални пат и пресек вектори. Во оваа докторска дисертација во Глава 5 ќе бидат предложени нови алгоритми за добивање на минимални пат вектори.

Да покажеме како се пресметува надежноста на повеќе – состојбен систем, кога минималните пат или минималните пресек вектори се познати.

Прво да претпоставиме дека минималните пат вектори за ниво d (MPV_d) се познати. Тогаш надежноста може да се пресмета на следниот начин:

$$M2TR_d = \sum_{h=1}^T P(\vec{x} \geq \vec{y}_h) - \sum_{h < k}^T P(\vec{x} \geq \vec{y}_h \wedge \vec{x} \geq \vec{y}_k) + \dots (-1)^T P(\vec{x} \geq \vec{y}_1 \wedge \dots \wedge \vec{x} \geq \vec{y}_T). \quad (3.2)$$

каде T е бројот од MPV_d (минимални пат вектори за ниво d) и $\vec{y}_h \in MPV_d$. Со користење на следнава нотација:

$$\max(\vec{z}_1, \dots, \vec{z}_s) = (\max(z_1^{(1)}, \dots, z_s^{(1)}), \dots, \max(z_1^{(l)}, \dots, z_s^{(l)})) \quad (3.3)$$

каде $z_u^{(v)}$ е v -та координата на координатата \vec{z}_u , равенката (3.3) може да биде запишана на следниот начин

$$M2TR_d = \sum_{h=1}^T P(\vec{x} \geq \vec{y}_h) - \sum_{h < k}^T P(\vec{x} \geq \max(\vec{y}_h, \vec{y}_k)) + \dots (-1)^T P(\vec{x} \geq \max(\vec{y}_1, \dots, \vec{y}_T)). \quad (3.4)$$

Ако минималните пресек вектори за ниво d (MCV_d) се познати, тогаш надежноста може да се пресмета на следниот начин:

$$M2TR_d = 1 - \sum_{h=1}^L P(\vec{x} \leq \vec{y}_h) + \sum_{h < k}^L P(\vec{x} \leq \vec{y}_h \wedge \vec{x} \leq \vec{y}_k) - \dots (-1)^L P(\vec{x} \leq \vec{y}_1 \wedge \dots \wedge \vec{x} \leq \vec{y}_T). \quad (3.5)$$

каде L е бројот од MCV_d (минимални пресек вектори за ниво d) и $\vec{y}_h \in MCV_d$. Со користење на следнава нотација:

$$\min(\vec{z}_1, \dots, \vec{z}_s) = (\min(z_1^{(1)}, \dots, z_s^{(1)}), \dots, \min(z_1^{(l)}, \dots, z_s^{(l)})) \quad (3.6)$$

каде $z_u^{(v)}$ е v -та координата на координатата \vec{z}_u , равенката (3.6) може да биде запишана како:

$$M2TR_d = 1 - \sum_{h=1}^L P(\vec{x} \leq \vec{y}_h) + \sum_{h < k}^L P(\vec{x} \leq \min(\vec{y}_h, \vec{y}_k)) - \dots (-1)^L P(\vec{x} \leq \min(\vec{y}_1, \dots, \vec{y}_T)). \quad (3.7)$$

Во програмскиот јазик C# е направена програма која врз база на најдените минималните пат вектори ја наоѓа надежноста. Програмата се базира на следниот алгоритам [46].

Алгоритам 3.1 - Алгоритам за пресметување на надежност на повеќе-состојбен транспортен систем $(G(V, E))$

Чекор 1. Се наоѓаат сите минимални пат вектори за ниво d , $d - MP = \{\vec{w}_1, \dots, \vec{w}_T\}$ за транспортниот систем $G(V, E)$,

Чекор 2. Се наоѓаат сите можни напразни подмножества од множеството $d - MP$,

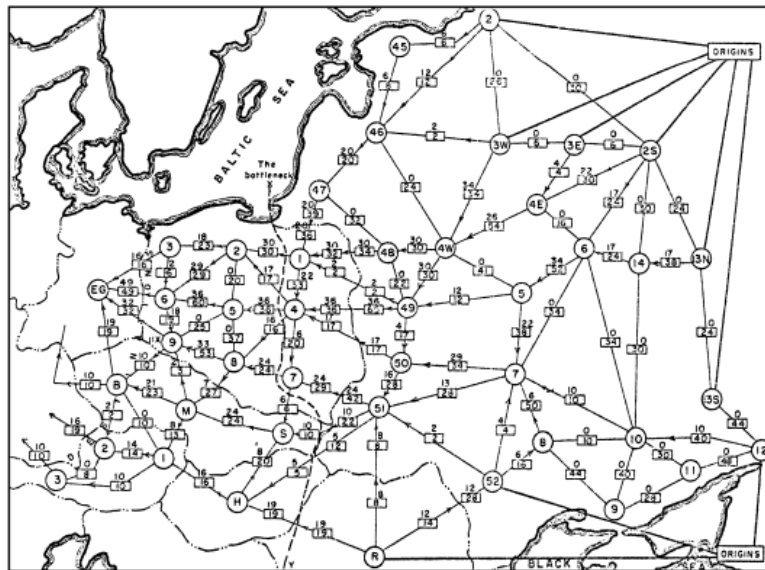
Чекор 3. За секое подмножество $d - MP$, $1 \leq r \leq 2^T - 1$ добиено во Чекор 2 се наоѓа супремумот, $\vec{v}_r = \sup MPV_{d_r}$, според подредувањето дадено во формула 3.3, на тој начин што секоја координата е еднаква на максимумот на соодветните координати на векторите од тоа подмножество.

Чекор 4. Се наоѓаат веројатностите $P(\varphi(\vec{x}) \geq \vec{v}_r)$ и се применува формулата 3.4 за пресметување на надежноста $M2TR_d$.

3.2 Проточни транспортни системи

Во средината на педесеттите години од минатиот век, научниците Т. Е. Harris and F. S. Ross од Air Force објавиле доверлива статија во која ја истражувале поврзаноста на железничката мрежа помеѓу Советскиот сојуз и неговите сателитски земји во Источна Европа. Мрежата била моделирана како граф со 44 јазли, кои ги претставуваат географските региони и 105 линкови кои ја претставуваат поврзаност на регионите со железничките мрежи. Секој линк има дадена тежина, што претставува количина на материјалот што може да биде пренесен од еден во друг регион. Всушност преку обиди и грешки, тие ја определиле максималната количина на материјали што може да биде пренесена од Русија во Европа, како и најевтиниот начин да се прекине мрежата со бришење (отстранување) на некои линкови, кое го нарекле “тесно грло”. Нивните резултати биле објавени дури во

1999. Овие резултати ја вклучувале и мапата на железничката мрежа на Варшавскиот пакт што е прикажана на Фигура 3.1 [18].



Фигура 3.1 Мапата на Haris и Ross на железничката мрежа на Варшавскиот пакт.

Ова се едни од првите забележани примени на проблемите за **максимален проток** и **минимален пресек**. За двата проблеми, транспортниот систем може да се претстави со насочен граф со два специјални јазли, извор и сливник. Низ транспортниот систем треба да биде пренесена одредена количина од некој материјал од изворот (каде што материјалот се произведува) до сливникот (каде што материјалот се консумира). Со помош на проточни транспортни системи може да бидат моделирани различни проблеми, како што се: проток на течности низ цевки, делови низ подвижна лента, струја низ електрични мрежи, информации низ комуникациски мрежи.

Секој насочен линк може да го замислиме како канал (цевка) низ кој поминува материјалот. Секој канал има почетен капацитет, што го претставува максималниот капацитет што може да биде пренесен низ тој канал. Јазлите може да ги замислиме како крстосници (спој) на каналите, и во сите јазли освен во изворот и сливникот материјалот само поминува низ јазлите без да се задржува во нив. Со други зборови, количината на материјал што влегува во јазолот мора да биде еднаква со количината на материјал што излегува од јазолот. Ова својство се нарекува “одржување на протокот” и е еквивалентно на кирховиот закон.

Во општ случај, кога разгледуваме **двотерминален проточен транспортен систем** $G(V, E)$ разгледуваме насочен граф со два специјални јазли извор s и сливник ($s \neq t$), во кој секој линк $(u, v) \in E$ има ненегативен **капацитет** $c(u, v) \geq 0$. Во овој труд се дава и теоријата за ненасочени транспортни системи, од причина што некои проблеми е пореално да се моделираат со нив. Сите дефиниции што ќе ги дадеме во оваа подглава се однесуваат само за насочени транспортни системи, додека во Глава 4 истите дефиниции ќе бидат модифицирани и за ненасочени транспортни системи. Функцијата c ќе ја наречеме функција од капацитети и проточниот транспортен систем ќе го означуваме со $G(V, E, c)$.

Проток во $G(V, E, c)$ е функција $f: E \rightarrow \mathbb{R}^+ \cup \{0\}$ што ги задоволува следниве две ограничувања:

1. *Ограничување на капацитетот:* $0 \leq f(u, v) \leq c(u, v)$, за секој $(u, v) \in E$, т.е. протокот на линкот не може да го надмине капацитетот на линкот.
2. *Одржување на протокот:* За сите $v \in V/\{s, t\}$

$$\sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w), \quad (3.8)$$

со други зборови, вкупниот проток што влегува во v е еднаков на вкупниот проток што излегува од v , за сите $v \in V/\{s, t\}$.

Претпоставуваме дека ако не постои линкот (u, v) т.е. $(u, v) \notin E$ тогаш $f(u, v) = 0$.

Вредноста на протокот $|f|$ се дефинира на следниот начин

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s), \quad (3.9)$$

т.е. вредноста на протокот е еднаква на вредноста кога од вкупниот проток што излегува од изворот ќе се одземе вкупниот проток што влегува во изворот. Вообичаено, кај проточните транспортни системи не постојат линкови кои влегуваат во изворот, и протокот што влегува во изворот ќе биде нула.

$$\sum_{v \in V} f(v, s) = 0,$$

Од друга страна, може да се покаже дека вредноста на протокот е еднаква на вредноста кога од вкупниот проток што влегува во сливникот се одзема вкупниот проток што излегува од сливникот.

Поради тоа што ограничувањето за одржување на проток важи за сите јазли, имаме:

$$\sum_{v \in V} \left(\sum_{w \in V} f(v, w) - \sum_{u \in V} f(u, v) \right) = \sum_{v \in V} \sum_{w \in V} f(v, w) - \sum_{v \in V} \sum_{u \in V} f(u, v) = 0,$$

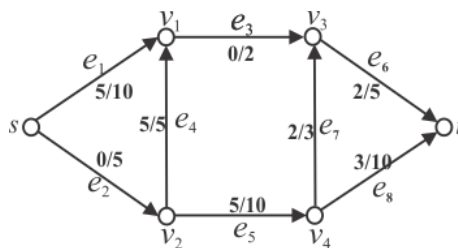
Од друга страна имаме дека

$$\begin{aligned} \sum_{v \in V} \left(\sum_{w \in V} f(v, w) - \sum_{u \in V} f(u, v) \right) &= \left(\sum_{w \in V} f(t, w) - \sum_{u \in V} f(u, s) \right) + \left(\sum_{w \in V} f(s, w) - \sum_{u \in V} f(u, t) \right) \\ &= |f| + \left(\sum_{w \in V} f(t, w) - \sum_{u \in V} f(u, t) \right). \end{aligned}$$

Од каде следува дека:

$$|f| = \left(\sum_{u \in V} f(u, t) - \sum_{w \in V} f(t, w) \right).$$

За протокот f велиме дека е **можен (feasible)** ако $f(u, v) \leq c(u, v)$, за секој $(u, v) \in E$. Најчесто, ќе разгледуваме само протоци кои се можни во однос на некоја фиксирана функција на капацитет c . Се вели протокот f го заситува (пренатрупува) линкот (u, v) ако $f(u, v) = c(u, v)$ и го поништува (анулира) линкот (u, v) ако $f(u, v) = 0$.



Фигура 3.2 Проток со вредност 5. Секој линк е означен со неговиот проток/капацитет.

Протокот е **максимален** ако има најголема можна вредност помеѓу сите протоци од изворот s до сливникот t во даден транспортен систем со дадена функција на капацитет.

Проблемот за наоѓање на максимален проток во даден проточен транспортен систем G со извор s и сливник t е да се најде протокот со најголема можна вредност, т.е. да се најде максималниот проток.

Псевдопроток е функција $f : E \rightarrow \mathbb{R}^+ \cup \{0\}$ дефинирана на линковите што го задоволуваат само ограничувањето на капацитет, и не е потребно линковите да го задоволуваат ограничувањето одржување на протокот. Јасно е дека секој проток е псевдопроток.

3.2.1 Алгоритам на Ford - Fulkerson

Алгоритамот на Ford - Fulkerson е алгоритам со кој се пресметува максималниот проток во транспортниот систем. Во литературата често се сретнува и како метод на Ford - Fulkerson бидејќи за овој алгоритам постојат повеќе имплементации со различно време на извршување. Пред да го дадеме алгоритамот на Ford - Fulkerson мора да ги воведеме поимите за остаточен транспортен систем, зголемувачки пат и минимален пресек [15].

Со алгоритамот на Ford - Fulkerson, вредноста на протокот се зголемува итеративно. Иницијалната вредност на протокот се зема да е еднаква на 0, т.е. $f(u, v) = 0$, за сите $(u, v) \in E$. Во секоја итерација, вредноста на протокот се зголемува со наоѓање на зголемувачки пат во остаточниот транспортен систем за протокот f , G_f . Бидејќи во секоја итерација на алгоритамот, се зголемува вредноста на протокот на транспортниот систем, треба да се води сметка дали ќе се промени (зголеми или намали) протокот на кој било линк во G . Намалувањето на протокот на некој линк може да биде потребно со цел да се овозможи алгоритамот да испрати поголем проток од изворот s до сливникот t . Оваа постапка се повторува се додека постојат зголемувачки патишта во соодветниот остаточен транспортен систем [15].

Сега да го дадемен алгоритмот на Ford - Fulkerson.

Алгоритам 3.2 - Алгоритам на Ford – Fulkerson ($G(V, E, c)$)

Чекор 1. Се иницијализира протокот f на 0, $f = 0$,

Чекор 2. Се додека постои зголемувачки пат P во остаточниот G_f се наоѓа проток f низ P ,

Чекор 3. Се враќа f .

3.2.2 Остаточен транспортен систем (residual network)

За даден проточен транспортен систем G и проток f , остаточниот транспортен систем се состои од линкови чии капацитети покажуваат за колку може да се промени протокот на линковите на G . На линковите на транспортниот систем се доделува додатен проток што е еднаков кога од капацитетот на линкот ќе се одземе протокот што е пуштен низ тој линк. Ако оваа вредност е позитивна, тогаш се нарекува остаточен капацитет $c_f(u, v) = c(u, v) - f(u, v)$. Во G_f се наоѓаат само оние линкови од G низ кои може да се пушти упте проток. Во спротивно доколку капацитетот на линкот е еднаков со протокот што веќе е пуштен низ линкот $c_f(u, v) = 0$, и овие линкови не се наоѓаат во G_f . Треба да се нагласи дека во G_f се содржат и линкови кои не се во G . Со цел да се претстави можното намалување на проток, се воведуваат линкови (v, u) во G_f чиј остаточен капацитет $c_f(u, v) = f(u, v)$.

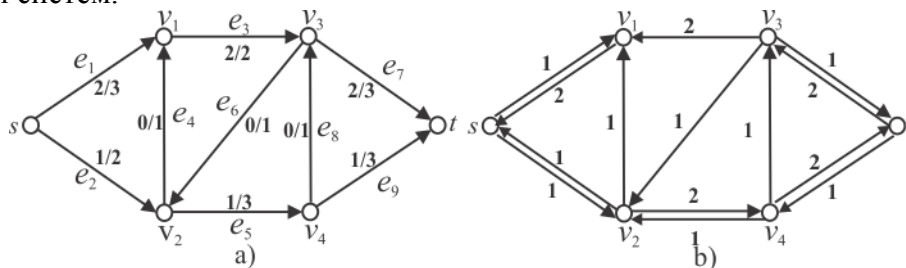
Да претпоставиме дека имаме проточен транспортен систем $G(V, E)$, со извор s и сливник t . Нека f е проток во транспортниот систем. Остаточниот капацитет се дефинира на следниот начин:

$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v), & \text{ако } (u, v) \in E \\ f(u, v), & \text{ако } (v, u) \in E \\ 0, & \text{инаку} \end{cases} \quad (3.10)$$

Остаточниот транспортен систем за транспортниот систем $G(V, E)$ индуциран од протокот f е $G_f(V, E_f)$ каде

$$E_f = \{(u, v) \in E : c_f(u, v) > 0\}. \quad (3.11)$$

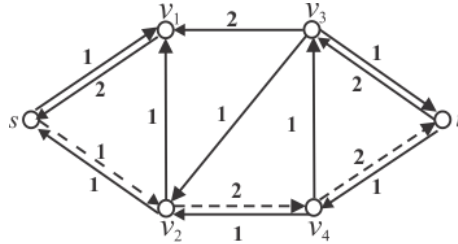
На Фигура 3.3 а) е претставен транспортен систем со проток f , кој има вредност $|f| = 3$. За секој линк е означено колкав е протокот што поминува низ линкот и кој е можниот капацитет на линкот (a/b). На Фигура 3.3 б) е претставен соодветниот остаточен транспортен систем.



Фигура 3.3 а) Проточна мрежа $G(V, E)$ со проток f , $|f| = 3$. б) Остаточен транспортен систем $G_f(V, E)$.

3.2.3 Зголемувачки пат

За даден проточен транспортен систем $G(V, E)$ и проток f , зголемувачки пат p е пат од s до t во остаточниот транспортен систем $G_f(V, E)$. Според дефиницијата на остаточен транспортен систем, ние може да го зголемуваме протокот на линкот (u, v) во зголемувачкиот пат p за $c_f(u, v)$, без да го нарушиме ограничувањето на капацитетот и на линкот (u, v) и на линкот (v, u) во транспортниот систем $G(V, E)$. На Фигура 3.4 со испрекината линија се означени линковите на зголемувачкиот пат $p = \langle s, v_2, v_4, t \rangle$.



Фигура 3.4 Остаточен транспортен систем $G_f(V, E)$ од Фигура 3.3 со зголемувачки пат $p = \langle s, v_2, v_4, t \rangle$.

За да не го нарушиме ограничувањето на капацитетот, ние може да го зголемиме протокот низ секој линк од патот $p = \langle s, v_2, v_4, t \rangle$ за 1, бидејќи најмалиот остаточен капацитет на овој пат е $c_f(s, v_2) = 1$. Најмалиот капацитет за кој може да се зголеми протокот на секој линк на патот p се нарекува остаточен капацитет на p .

$$c_f(p) = \min\{c_f(u, v) : (u, v) \text{ е линк во } p\}. \quad (3.12)$$

3.2.4 Пресеци во проточни мрежи

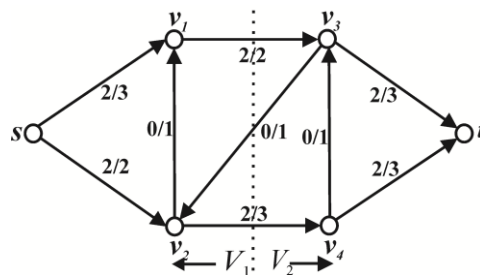
Со алгоритмот на Ford - Fulkerson повеќе пати се врши зголемување на протокот сè додека не се најде максималниот проток. Се поставува прашањето како ќе знаеме дека алгоритмот завршува? Теоремата за максимален проток – минимален пресек кажува дека протокот е максимален ако и само ако неговиот остаточен транспортен систем не содржи повеќе зголемувачки патишта. За доказ на оваа теорема потребно е да се воведат поимот пресек во проточен транспортен систем.

Пресек (V_1, V_2) на проточната мрежа $G(V, E)$ е поделба на множеството од јазли V во множества V_1 и $V_2 = V / V_1$, $s \in V_1$ и $t \in V_2$. Ако f е проток, тогаш протокот низ пресекот $f(V_1, V_2)$ се дефинира на следниот начин:

$$f(V_1, V_2) = \sum_{u \in V_1} \sum_{v \in V_2} f(u, v) - \sum_{u \in V_1} \sum_{v \in V_2} f(v, u). \quad (3.13)$$

Капацитетот на пресекот (V_1, V_2) е

$$c(V_1, V_2) = \sum_{u \in V_1} \sum_{v \in V_2} c(u, v). \quad (3.14)$$



Фигура 3.5 Пресек во проточниот транспортен систем од Фигура 3.3 а) каде $V_1 = \{s, v_1, v_2\}$ и $V_2 = \{v_3, v_4, t\}$. Протокот $f(V_1, V_2) = 4$ и капацитетот $c(V_1, V_2) = 5$.

Минимален пресек во транспортниот систем е пресек чиј што капацитет е помал од капацитетот на сите други пресеци на мрежата.

Лема 3.1 Нека f е проток во проточен транспортен систем G со извор s и сливник t , и нека (V_1, V_2) е кој било пресек во транспортниот систем. Тогаш протокот низ пресекот $f(V_1, V_2) = |f|$.

Следната последица произлегува од Лема 3.1

Последица 3.1 Вредноста на секој проток f во проточниот транспортен систем G е ограничена од горе со капацитетот на кој било пресек на G . Значи протокот во транспортниот систем не може да биде поголем од капацитетот на кој било пресек во транспортниот систем.

Следната теорема ни дава начин како да го пресметаме максималниот проток во транспортниот систем. Веќе кажавме дека ако остаточниот транспортен систем $G_f(V, E)$ не содржи зголемувачки пат тогаш протокот на транспортниот систем е максимален. Од следната теорема може да заклучиме дека доколку вредноста на некој проток е еднаква на капацитетот на некој пресек, тогаш тој проток е максимален.

Теорема 3.1 (Теорема за максимален проток – минимален пресек)

Ако f е проток во проточниот транспортен систем $G(V, E)$ со извор s и сливник t , тогаш следниве услови се еквивалентни:

1. f е максимален проток во G ,
2. Остаточниот транспортен систем G_f не содржи зголемувачки пат,
3. $|f| = c(V_1, V_2)$ за кој било пресек (V_1, V_2) на G .

Откако ги воведовме поимите за остаточен транспортен систем, зголемувачки пат и остаточен капацитет, сега ќе дадеме едно едноставно проширување на псевдокодот за алгоритмот што го дадовме погоре:

Алгоритам 3.3 - Алгоритам на Ford – Fulkerson ($G(V, E, c)$)

Чекор 1. За секој линк $(u, v) \in G, f(u, v) = 0$,

Чекор 2. Сè додека постои зголемувачки пат \mathcal{P} од s до t во остаточниот транспортен систем G_f ,

Чекор 3. $c_f(\mathcal{P}) = \min\{c_f(u, v) \mid (u, v) \in \mathcal{P}\}$

Чекор 4. За секој линк $(u, v) \in \mathcal{P}$

Чекор 5. Ако $(u, v) \in E$

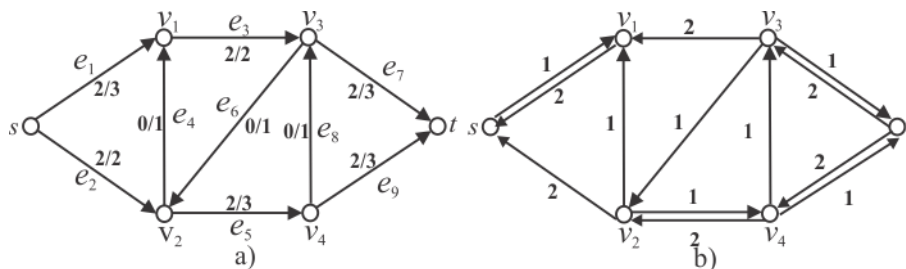
$$f(u, v) = f(u, v) + c_f(\mathcal{P}),$$

инаку

$$f(v, u) = f(v, u) - c_f(\mathcal{P}),$$

Сложеноста на овој алгоритам е $O(|f^*||E|)$ каде $|f^*|$ е вредноста на максималниот проток во проточниот транспортен систем.

Пример 3.1 Да го најдеме максималниот проток на транспортниот систем даден на Фигура 3.3 а). На Фигура 3.3 б) е даден остаточниот транспортен систем за протокот со вредност 3. Според чекор 3 од алгоритмот, протокот на сите линкови од патот $\mathcal{P} = \langle s, v_2, v_4, t \rangle$ ќе се зголеми за 1 бидејќи линковите (s, v_2) , (v_2, v_4) и (v_4, t) се елементи од E .



Фигура 3.6 а) Проточен транспортен систем $G(V, E)$ со проток $f, |f| = 4$. б) Остаточен транспортен систем $G_f(V, E)$

Можеме да забележиме дека остаточниот транспортен систем прикажан на Фигура 3.6 б) не содржи зголемувачки пат, од каде според теоремата за за максимален проток – минимален пресек може да заклучиме дека максималната вредност на протокот на транспортниот систем од Фигура 3.3 а) е 4.

3.3. Преглед на истражувањата на надежноста на двотерминални повеќе-состојбени транспортни системи

Во овој дел прво ќе дадеме краток осврт на трудовите кои се однесуваат на пресметување на надежност на двотерминални транспортни системи. Во Прилозите А и Б подетално ќе ги разработиме алгоритмите предложени од [51], [34] и [36].

Во [62] се разгледуваат стохастички транспортни системи за снабдување со електрична енергија. Во трудот е предложен генетски алгоритам што се базира на методот

на интеграција (спојување) на минималните патишта и рекурзивните суми од дисјунктни производи. Целта на предложениот алгоритам е да се најде максималната надежност на транспортниот систем за пренос на електрична енергија со оптимални линии (вододи, линкови) за пренос. Авторите разгледуваат транспортни системи кои имаат повеќе – состојбени компоненти. Бидејќи во трудот се разгледуваат транспортни системи за пренос на електрична енергија, повеќе-состојбената надежност за ниво d се дефинира како веројатност дека d единици може успешно да бидат пренесени од изворот (генератор за електрична енергија) до крајните дестинации (корисниците). За да се покаже ефикасноста на алгоритмот се разгледуваат и едноставен транспортен систем за пренос на електрична енергија и покомплексен (реален) транспортен систем за пренос на електрична енергија.

Во трудот [8] е предложен алгоритам со кој се добиваат сите минимални пресеци за ниво d , за повеќе-состојбен проточен транспортен систем. Вредностите на капацитетите на линковите се ненегативни цели броеви. Веројатноста да се достигне максимален проток за различни нивоа на работа на транспортниот систем, како и пресметување на неговата надежност, може да се изврши со помош на K -решетки (делумно подредени множества во кои секој два елементи имаат супремум и инфимум) кои се генерираат од сите подмножества од фамилијата на сите минимални пресек множества (бинарни минимални пресек множества). Меѓутоа, големината на ваквата фамилија од минимални пресек множества е $2^m - 1$ (каде m е бројот на минимални пресек множества) и расте експоненцијално по m . Таков проточен транспортен систем може да се разгледува како повеќе - состојбен систем со повеќе - состојбени компоненти поради што неговата надежност може да се оцени во термини на горно гранични точки за секое ниво d (наречени овде минимални пресек множества за ниво d). Во трудот е даден алгоритам со кој се генерираат сите минимални пресек множества за ниво d за секое минимално пресек множество и за секое ниво на капацитет на системот d . Дадени се и примери со кои се покажува како се генерираат сите минимални пресек множества за ниво d и за едниот од примерите се пресметува надежноста.

Во [56] е предложен алгоритам за наоѓање на сите минимални пресеци за ниво d во ограничен проточен транспортен систем $G(V, E, c)$ кој работи единствено во согласност со дефиницијата за минимални пресеци за ниво d , каде $c(e)$ е максималниот капацитет на линкот e . Во трудот се претпоставува дека минималните пресеци (бинарните минимални пресеци) се познати однапред. Исто така, направена е и споредба на предложениот алгоритам со други познати алгоритми во литературата. Авторот покажува дека овој алгоритам е поефикасен од другите познати алгоритми. Со анализирање на алгоритмот и со споредба со другите познати алгоритми покажано е дека предложениот метод е лесен за разбирање и имплементација. Пресметана е и комплексноста на алгоритмот и извршена е споредба со комплексноста на другите познати алгоритми. Меѓутоа, Н. Salehi-Fathabadi и М. Forghani-elahabad, во трудот [14] покажале дека некои минимални пресеци не се добиваат со овој алгоритам. Тие установиле дека грешката се јавува во дефиницијата на множеството E_{ij} , поточно во Чекор 2 од алгоритмот на Yeh. Прво да го видиме псевдокодот на алгоритмот, па да кажеме како се јавува грешката во овој алгоритам.

Алгоритам на *Yeh* - (Минимални пресеци C_i за тежински граф $G(V, E, c)$)

Чекор 1. Нека $j \leftarrow 0$, $V_i \leftarrow \emptyset$, $V_k \leftarrow \emptyset$.

Чекор 2. Нека $j \leftarrow j + 1$ и се користи имплицитен алгоритам за наоѓање на сите кандидати за минимални пресеци за ниво d , со примена на следниот математички модел.

$\sum f(e) = d$ за сите $e \in C_i$ (каде $f(e)$ е протокот низ линкот e),

$C_i(e) \leq c(e)$ за сите $e \in C_i$,

$C_i(e) = c(e)$ за сите $e \notin C_i$.

Ако не постои кандидат за минимален пресек за ниво d алгоритмот завршува.

Во спротивно $V_i = V_i \cup \{C_{ij}\}$.

Чекор 3. Ако $M(C_{ij}) = d$ тогаш $E_{ij} \leftarrow C_i$, во спротивно C_{ij} не е d -МС и оди на Чекор 1.

(Каде $M(\vec{X})$ е максималниот проток од изворот s до сливникот t , кога системот е во состојба \vec{X}).

Чекор 4. Ако $E_{ij} = \emptyset$ тогаш C_{ij} е d -МС, $V_k = V_k \cup \{C_{ij}\}$ и оди на Чекор 1. Во спротивно избери кој било линк e од E_{ij} и нека $E_{ij} \leftarrow E_{ij} \setminus \{e\}$.

Чекор 5. Ако $M(C_{ij} + o(e)) = d$ тогаш C_{ij} не е d -МС и оди на Чекор 1, во спротивно оди на Чекор 3.

Значи, како што кажавме грешката на алгоритмот на *Yeh* се случува во Чекор 3, бидејќи во дефиницијата на E_{ij} се вклучени и линкови за кои $C_{ij}(e) = c(e)$, што во Чекор 5 резултира со отфрлање на некои вистински минимални пресеци за ниво d , бидејќи ако линкот e во C_{ij} има максимален капацитет, јасно дека за $C_{ij} + o(e)$ ќе се добие состојба на транспортниот систем во која линкот e ќе има поголем капацитет од неговиот максимален капацитет.

Поради ова во алгоритмот на Н. Salehi-Fathabadi и М. Forghani-elahabad E_{ij} се дефинира на следниот начин:

$$E_{ij} = \{e \mid e \in C_i \wedge C_{ij}(e) < c(e)\}.$$

Во [57] е предложен алгоритам за наоѓање на сите минимални пресеци за ниво d во повеќе-состојбен проточен транспортен систем, под претпоставка дека минималните пресеци (бинарните минимални пресеци) се познати однапред. Авторите тврдат дека предност на овој алгоритам е што не се добиваат исти минимални пресеци за ниво d повеќе од еднаш. Ова е значајно, бидејќи не е потребно да се прават дополнителни пресметки за елиминирање на минималните пресеци што се јавуваат повеќе од еднаш. Предложениот алгоритам има временска сложеност $O(mp^2\sigma)$ каде p е број на минимални пресеци, σ е број на минимални пресеци за ниво d , и m е број на линкови.

Во трудот [63] се дискутираат три начини за подобрување на ефикасноста за решавање на d -МС проблемот. Првиот начин ги користи својствата на протокот во транспортниот систем. Специјален транспортен систем “Network with Joint Parallel Part” (Транспортен систем со заеднички паралелен дел) се разгледува. Овој транспортен систем многу почесто се јавува во пракса отколку добро познатиот сериско-паралелен

транспортен систем. Вториот начин е со редуцирање на бројот на кандидати за минимални пресеци за ниво d . Со третиот начин ефикасно се наоѓаат минималните пресеци за ниво d што се појавуваат повеќе од еднаш. Исто така, во овој труд е модифициран и алгоритамот предложен од Yeh [58]. Се предлага нов модифициран алгоритам во кој се додава на нов чекор, пред-пресметувачки чекор (pre-numerating step). Истакнато е дека овој чекор може да се користи за добивање на минимални пресек вектори во транспортни системи со хетерогени компоненти.

Во трудот [16] е предложен алгоритам за пресметување на бројот на минимални пресеци во тежински планарен граф. Сложеноста на алгоритамот е $O(nd+n \log n)$ каде n е бројот на јазли, и d е должината на најкраткиот пат од s до t во соодветниот нетежински граф. Недостаток на алгоритамот е што работи само за планарни графови. Предложениот алгоритам се базира на алгоритмот на Ford-Fulkerson. Прво се пресметува максималниот протокот s до t , така што насочените линкови да не формираат циклуси. Потоа се формира остаточениот граф и од него се формира нов граф, такашто секоја цврсто поврзана компонента се заменува со јазол. Од овој граф се формира остаточен нетежински планарен граф. Со користење на некои алгоритми за насочени ациклични графови се наоѓа бројот на минимални пресеци. Резултатите добиени во овој труд имаат директна примена во сегментацијата на слики и во други проблеми од компјутерската визуелизација.

Во трудот [17] се разгледува проблемот за наоѓање на сите минимални пресеци. проблемот за наоѓање на сите минимални пресеци може да се реши со користење на кој било од алгоритмите за максимален проток и со користење на теоремата за максимален проток - минимален пресек на Ford - Fulkerson. Со секој максимален проток е поврзана бинарна релација и минималните пресеци се идентификуваат со затворачите на оваа релација. Се наоѓаат строго поврзани компоненти и се формира нов граф на тој начин што секоја ваква компонента се заменува со јазол. Резултатите од овој труд имаат примена кај параметарската анализа на мрежите, кај поставувањето на јазлите во мрежите, кај проблемот за наоѓање на максимален затворач и кај псеудо булови оптимизации.

Во трудот [32] е предложен рекурзивен алгоритам којшто се базира на принципот на сума од дисјунктни производи за оценување на надежноста на повеќе-состојбен транспортен систем. Во алгоритамот се претпоставува дека минималните пат вектори за ниво d се дадени. Работата на овој алгоритам е демонстрирана со разгледување на пример. Преку споредба со други познати алгоритми е докажана ефикасноста на предложениот алгоритам. Алгоритамот е многу поефикасен кога бројот на компоненти на повеќе-состојбен транспортен систем е голем и може да се користи за оценување на надежноста на комплексни повеќе-состојбени транспортни системи со голем број на компоненти.

Во трудот [64] надежноста се пресметува со користење на минимални пат вектори и рекурзивни суми од дисјунктни производи. Главната цел на трудот е да се оцени надежноста на електронски пренос на трансакции што се врши од страна на комерцијалните банки. Структурата на мрежата на банката како и веројатноста на успехот на пренос на трансакции се добиени со собирање на реални податоци. Разгледан е и пример како работи комерцијална банка во Тајван.

Во трудот [22] е предложена методологија што го решава проблемот за наоѓање на сите минимални пресек вектори за повеќе-состојбен двотерминален транспортен систем. Предноста на овој алгоритам е што не бара какви било ограничувања за вредностите што може да ги примаат капацитетите на линковите. Но да истакнеме дека во овој алгоритам постои грешка и некои минимални пресек вектори не се добиваат. проблемот се случува во Чекор 1 од алгоритамот кога се бришат комбинации од вредности на капацитетите што се доминантни (се поголеми од) над други вредности од капацитети. Во тој чекор се добиваат минималните пресек вектори за транспортниот подсистем. проблемот се појавува поради тоа што не сите минимални пресек вектори за целиот транспортен систем се и минимални пресек вектори за транспортните подсистеми. Овој алгоритам е детално разработен и објаснето е како се јавува грешка и демонстрирано е со пример во [46].

Во трудот [19] е предложен алгоритам за добивање на минимални пат вектори, што може да се примени за релативно мали транспортни системи. Со овој алгоритам се добиваат кандидати за минимални пресек вектори што не се минимални, од каде следува дека треба да се извршуваат додатни пресметки за да се елиминираат овие кандидати за минимални пресек вектори. Вектори се елиминираат со заемна споредба, но оваа процедура е релативно скапа, бидејќи бројот на минимални пат вектори е многу поголем од бројот на линкови и бројот на јазли во мрежата.

Во трудот [45] е предложен алгоритам за наоѓање на минимални пат вектори за повеќе-состојбен двотерминален транспортен систем. Главната предност на овој алгоритам е што за вредност на капацитетите на линковите може да прима кој било реален број. И со овој алгоритам се добиваат исти минимални пат вектори повеќе од еднаш и кандидати за минимални пат вектори што не се минимални. Овие вектори се елиминираат со заемна споредба.

Во трудот [51] е предложен алгоритам за наоѓање на минимални пат вектори во проточни мрежи, главно со користење на законот за одржување на проток. Овој алгоритам и алгоритамот предложен од M. Forghani – elahabad и N. Mahdavi-Amiri [34] ќе бидат разгледани во Прилог А.

Во трудот [36] се предложени алгоритми за наоѓање на минималните пат вектори за сите нивоа. Сите овие алгоритми се едноставни за разбирање и се базираат на едноставни операции помеѓу вектори. Овој алгоритам детално ќе биде разгледан во ПрилогБ.

Глава 4

4. Врска помеѓу минимален пат вектор и проток кај двотерминални проточни транспортни системи

Надежноста на повеќе - состојбени транспортни системи може да се пресмета ако се познати минималните пат вектори. Во литературата што ја разгледувавме, во најголем број се добиваат кандидати за минимални пат вектори кои не се минимални. Исто така видовме дека овие кандидати се елиминираат со заемно споредување. Поточно, ги споредуваме сите кандидати за минимални пат вектори и доколку за некој кандидат за минимален пат вектор постои друг кандидат, што е помал од него тогаш тој се елиминира. Оваа процедура за заемна споредба помеѓу векторите е релативно скапа, бидејќи бројот на минимални пат вектори вообичаено е поголем од бројот на јазли и линкови. Исто така, заклучивме дека еден кандидат за минимален пат вектор не е минимален пат вектор ако во графот што се формира од овој вектор постои циклус. Меѓутоа, тоа го заклучивме интуитивно, и мислевме дека е очигледно. Со тек на време, работејќи на оваа проблематика, увидовме дека тоа не е тривијално, и дека е потребна подетална анализа и разработка на истото. Исто така, видовме и во литературата од оваа област дека во најголем број од трудовите се разгледуваат графови без циклуси, меѓутоа не се кажува зошто се користат токму овие графови, и никаде не е даден доказ дека графот што го формира минималниот пат вектор е ацикличен. Ацикличен граф е оној граф што не содржи циклуси. За таа цел, ние прво ги разгледавме проточните мрежи и ја најдовме врска помеѓу минималните пат вектори и протоците.

Во продолжение на оваа глава ќе ги анализираме својства на минималните пат вектори за ниво d и ќе ја покажеме врска помеѓу минималните пат вектори за ниво d и протоците за ниво за d за двотерминален проточен транспортен систем.

Во првата подглава ги анализираме својства на минималните пат вектори за ниво d , и ќе ја покажеме врска помеѓу минималните пат вектори за ниво d и протоци за ниво за d на двотерминалните насочени транспортни системи. Тоа ни помага да развиеме

стратегија за проверување дали некој кандидат за минимален пат вектор за ниво d е минимален пат вектор со временска комплексност $O(|E|)$, што е значајно подобро од $O(|V|^2|E|^{3/2})$, што е временска комплексност на стратегија дадена во [34]. Покрај тоа, со користење на понатамошна анализа во Глава 5 ќе ја најдеме врската помеѓу два минимални пат вектори и ќе предложиме нови алгоритми со кои директно се наоѓаат сите минимални пат вектори за ниво d . Ќе ги објаснеме и предностите на овој пристап, посебно кога d е максимално ниво.

Вообичаено е двотерминалните проточни транспортни системи да се дефинираат преку насочени графови, но постојат голем број на примени каде е многу попрочно да се користат ненасочени графови. Во втората подглава ја анализираме врската помеѓу минималните пат вектори и протоците кај ненасочени транспортни системи.

4.1 Врска помеѓу минималните пат вектори и функциите на проток кај двотерминални проточни насочени транспортни системи

Во ова поглавје ќе ја анализираме врската помеѓу минималните пат вектори и протоци за насочени транспортни системи [41].

Разгледуваме двотерминален насочен проточен транспортен систем $G(V, E)$, со два специјални јазли извор s и сливник t ($s \neq t$), во кој секој линк $(u, v) \in E$ има ненегативен капацитет $c(u, v) \geq 0$. Функцијата c се нарекува функција од капацитет. Како што веќе кажавме во Глава 3, овие транспортни системи ќе ги означуваме со $G(V, E, c)$.

Дефинициите за проток, псевдопроток, максимален проток во транспортен систем, како и законите за одржување на проток и ограничување на капацитет се дадени во Глава 3.

Овде ќе дадеме некои дефиниции кои се потребни за анализа на врската помеѓу минималните пат вектори и протоците.

Да претпоставиме дека множеството од линкови во проточниот транспортен систем е подредено, т.е. $E = \{e_1, e_2, \dots, e_{|E|}\}$. Ако ги разгледуваме линковите како компоненти, транспортниот систем претставува систем со повеќе – состојбени компоненти. Може да се претпостави дека секоја компонента, секој линк e_i , може да работи со некое барано ниво $x_i \leq c(e_i)$. Нека векторот \vec{x} е вектор од состојби. Во теоријата на повеќе - состојбена надежност [6], како што веќе дефинираме во Глава 2, векторот \vec{x} се нарекува пат вектор за ниво d ако и само ако системот во состојба \vec{x} работи со ниво еднакво или поголемо од d .

Наредно што ќе направиме е воведување на некои дефиниции што ја дават врската помеѓу системите со повеќе-состојбени компоненти и двотерминалните транспортни системи.

Дефиниција 4.1 Нека $G(V, E, c)$ е двотерминален проточен транспортен систем. За псевдопроток l_c , дефинираме **вектор од состојби** \vec{x}_{l_c} **индуциран од** l_c со

$$x_i = l_c(e_i). \quad (4.1)$$

За секој вектор \vec{x} , со $x_i \leq c(e_i)$, дефинираме **псевдопроток** $l_{\vec{x}}$ **индуциран од** \vec{x} , со

$$l_{\vec{x}}(e_i) = x_i. \quad (4.2)$$

Векторот од состојби \vec{x} ќе го наречеме **вектор индуциран од проток (flow vector)**, кога $l_{\vec{x}}$ е проток.

Aggarwal et al. [27] ја дефинираат двотерминалната надежност како веројатност дека во транспортниот систем може да се пренесе бараниот проток од изворот до сливникот. Со други зборови, системот е во работна состојба ако и само ако е можно успешно да се пренесе бараниот проток од изворот до сливникот. Следниве дефиниции го објаснуваат ова многу попрецизно.

Дефиниција 4.2 Нека $G(V, E, c)$ е двотерминален проточен транспортен систем и \vec{x}_{l_c} е вектор од состојби индуциран од псевдопротокот l_c . Ќе велиме дека \vec{x}_{l_c} е пат вектор за ниво d , ($d - P$), ако и само ако проток d може да биде пренесен во двотерминалниот транспортен систем $G(V, E, l_c)$. Векторот од состојби \vec{x} е минимален пат вектор за ниво d , ($d - MP$), ако и само ако двотерминалниот проточен транспортен систем $G(V, E, l_{\vec{x}})$ има максимален проток d , и за секој друг вектор $\vec{x}' \leq \vec{x}$, двотерминалниот транспортен систем $G(V, E, l_{\vec{x}'})$ има максимален проток помал од d .

Дефинициите за остаточен капацитет, остаточен транспортен систем и зголемувачки пат се дадени во Глава 3. Овде можеме да нагласиме дека секој проточен транспортен систем $G(V, E)$ може да се разгледува и како остаточен транспортен систем $G_f(V, E_f)$ индуцирана од протокот f , каде $f(u, v) = 0$ за сите (u, v) . На истиот начин како што дефиниравме зголемувачки пат може да дефинираме и **зголемувачки циклус** како прост циклус \mathcal{C} од некој јазол v до v во остаточниот транспортен систем G_f . За секој зголемувачки циклус \mathcal{C} во остаточниот транспортен систем G_f , дефинираме **зголемувачки вектор** за ниво d' за циклусот \mathcal{C} , $\vec{y}^{\mathcal{C}, d'}$, со

$$y_i^{\mathcal{C}, d'} = \begin{cases} d', & \text{ако } e_i = (u, v) \in E \text{ и } (u, v) \text{ е во } \mathcal{C}, \\ -d', & \text{ако } e_i = (u, v) \in E \text{ и } (v, u) \text{ е во } \mathcal{C}, \\ 0, & \text{инаку.} \end{cases} \quad (4.3)$$

за некое $d' \leq \min\{c_f(e_i) \mid e_i \in \mathcal{C}\}$.

Како што нагласивме, сакаме да ја најдеме врската помеѓу минималните пат вектори за ниво d и протоците за истото ниво. За таа цел, сега ќе дадеме пристап со помош на кој ќе проверуваме дали некој даден проток за ниво d одговара на минимален пат вектор за ниво d .

Даден е проток f , со E^f го означуваме множеството од сите линкови со позитивен проток, т.е. $E^f = \{e \in E \mid f(e) > 0\}$. Нетежинскиот граф $G(V, E^f)$ го нарекуваме **граф индуциран од f** .

За да провериме дали некој проток одговара на минимален пат вектор треба да провериме дали графот $G(V, E^f)$ индуциран од f е ацикличен. Ова е прикажано со следната теорема и илустрирано со примерот даден по теоремата.

Теорема 4.1 Векторот од состојби \vec{x} е минимален пат вектор за ниво d (d -МП) за двотерминалниот проточен транспортен систем $G(V, E, c)$ ако и само ако псевдопротокот $l_{\vec{x}}$ е проток со вредност $|l_{\vec{x}}| = d$, и соодветниот граф $G(V, E^{l_{\vec{x}}})$ индуциран од $l_{\vec{x}}$ е ацикличен.

Доказ: Претпоставуваме дека \vec{x} е d -МП. Прво ќе докажеме дека $l_{\vec{x}}$ е проток со $|l_{\vec{x}}| = d$. Бидејќи \vec{x} е d -P, максималниот проток на $G(V, E, l_{\vec{x}})$ е еднаков на d . Тогаш имаме проток f за ниво d за $G(V, E, l_{\vec{x}})$. Нека \vec{y}_f е вектор индуциран од f . Јасно е дека \vec{y}_f е d -P и $\vec{y}_f \leq \vec{x}$. Бидејќи не постои помал пат вектор за ниво d од \vec{x} , имаме $\vec{y}_f = \vec{x}$, од каде следува дека $l_{\vec{x}} = f$. Со ова е докажано дека $l_{\vec{x}}$ е проток со $|l_{\vec{x}}| = d$.

Останува да се докаже дека графот $G(V, E^{l_{\vec{x}}})$ е ацикличен. За таа цел ќе го претпоставиме спротивното т.е. дека во графот $G(V, E^{l_{\vec{x}}})$ има циклус. Нека $G(V, E^{l_{\vec{x}}})$ има прост циклус, и $e_{i_1}, e_{i_2}, \dots, e_{i_r}$, се линковите кои припаѓаат на тој циклус. Со $Y = \{i_1, \dots, i_r\}$ ќе го означиме множеството од индексите на линковите во циклусот. Нека $m = \min\{x_j \mid j \in Y\}$ и \vec{y} е дефинирано на следниот начин: $y_j = \begin{cases} m, & j \in Y \\ 0, & j \notin Y \end{cases}$. Ќе покажеме дека векторот од состојби $\vec{z} = \vec{x} - \vec{y}$ е пат вектор за ниво d , d -P.

Прво ќе покажеме дека вкупниот проток за секој јазол u останува ист. Јасно, ако u не припаѓа во циклусот, вкупниот проток што влегува и излегува во јазолот не се променува. Ако u припаѓа во циклусот, имаме:

$$\sum_{v \in V} l_{\vec{z}}(u, v) - \sum_{v \in V} l_{\vec{z}}(v, u) = \sum_{v \in V} l_{\vec{x}}(u, v) - m - \sum_{v \in V} l_{\vec{x}}(v, u) + m = \sum_{v \in V} l_{\vec{x}}(u, v) - \sum_{v \in V} l_{\vec{x}}(v, u), \quad (4.4)$$

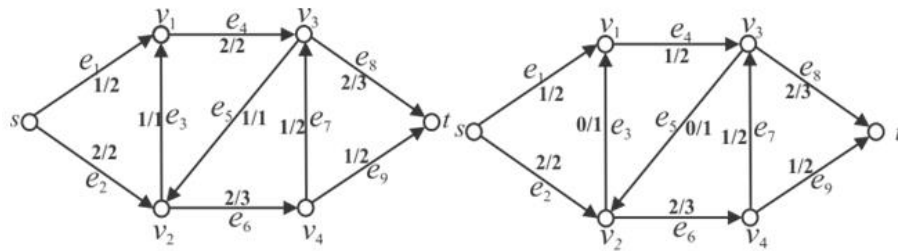
па протокот повторно останува ист.

Значи, ограничувањето за одржување на проток е задоволено и $|l_{\vec{z}}| = |l_{\vec{x}}| = d$ па \vec{z} е пат вектор за ниво d . Бидејќи $c(e_i) > x_i - y_i > 0$, имаме $0 \leq \vec{x} - \vec{y} < \vec{x}$. Пат векторот $\vec{z} = \vec{x} - \vec{y}$ е помал од векторот \vec{x} , што е во контрадикција со претпоставката дека \vec{x} е d -МП.

За да го покажеме спротивното, претпоставуваме дека векторот од состојби \vec{x} кој го индуцира протокот $l_{\vec{x}}$ со вредност $|l_{\vec{x}}| = d$ и графот што е индуциран од $l_{\vec{x}}, G(V, E^{l_{\vec{x}}})$ е ацикличен. За таа цел ќе покажеме дека секој вектор од состојби $\vec{x}' < \vec{x}$, е индуциран од проток со вредност $|l_{\vec{x}'}| < d$.

Нека претпоставиме дека постои пат вектор \vec{x}' за ниво d така што $\vec{x}' < \vec{x}$. Без губење на општост, можеме да претпоставиме дека $(\exists! i) \vec{x}' < \vec{x}$. Нека $e_i = (w, w_1)$. Имаме дека $x_i = l_{\vec{x}}(w, w_1) > l_{\vec{x}'}(w, w_1) = x'_i$, и за сите други јазли u и v , $l_{\vec{x}}(u, v) > l_{\vec{x}'}(u, v)$. Бидејќи $G(V, E^{l_{\vec{x}}})$ е ацикличен, јазлите може тополошки да се сортираат. Истото тополошко сортирање може да се примени и за $G(V, E^{l_{\vec{x}'}})$. Ако земеме V_1 да биде множеството од сите јазли помеѓу s и w , вклучувајќи ги и s и w , и $V_2 = V \setminus V_1$, ќе добиеме пресек во $G(V, E^{l_{\vec{x}'}})$ со проток $d - (x_i - x'_i) < d$. Со ова е докажано дека $G(V, E^{l_{\vec{x}'}})$ има максимален проток помал од d , што е контрадикција со претпоставката дека \vec{x}' е d - P . \square

Теорема 4.1 е илустрирана на Фигура 4.1. Протокот прикажан на Фигура 4.1 а) е проток за ниво 3, но векторот од состојби индуциран од него не е минимален пат вектор за ниво 3. Забележуваме дека векторот од состојби индуциран од протокот за ниво 3 прикажан на Фигура 4.1 б) индуцира вектор од состојби што е помал од векторот од состојби што е индуциран од протокот прикажан на Фигура 4.1 а). Протокот во Фигура 4.1 а) го содржи циклусот $\langle v_1, v_3, v_2, v_1 \rangle$, додека протокот во Фигура 4.1 б) нема циклус. Оттука векторот $(1, 2, 1, 2, 1, 2, 1, 2, 1)$ не е 3- MP , додека векторот $(1, 2, 0, 1, 0, 2, 1, 2, 1)$ е 3- MP .



Фигура 4.1 а) Проток за ниво 3 со $(1, 2, 1, 2, 1, 2, 1, 2, 1)$ како вектор индуциран од неа, кој не е 3- MP ; б) Проток за ниво 3 со $(1, 2, 0, 1, 0, 2, 1, 2, 1)$ кој е 3- MP .

4.2 Врска помеѓу минималните пат вектори и функциите на проток кај двотерминални проточни ненасочени транспортни системи

Најголем број од алгоритмите за наоѓање на минимални пат вектори или минимални пресек вектори се наменети за насочени транспортни системи. Од друга страна пак, постојат голем број на системи како телекомуникациски системи, транспортни системи, компјутерски мрежи и други кои е природно да се моделираат со ненасочени мрежи, со оглед на тоа дека линковите во овие мрежи се користат во двете насоки [2]. Во такви случаи линкот помеѓу две дестинации не може лесно да се замени со два насочени линкови со исти капацитети. На пример, ако една цевка во системот за дистрибуција на вода не функционира, тогаш оваа цевка не е во употреба во двете насоки, додека ако еден правец на автопатот е затворен, другиот правец сè уште може да се користи [2]. Затоа нашето внимание во ова подглава ќе биде насочено на анализирање на минималните пат вектори во ненасочени транспортни системи. Поточно, во ова поглавје ќе ја дадеме врска помеѓу минимални патишта за ниво d и протокот со вредност d за ненасочени транспортни системи при што ќе го користиме истиот пристап што го користевме и кај насочените транспортни системи во претходната подглава.

Прво ќе докажеме дека секој проток во ненасочен двотерминален транспортен систем дефинира единствен насочен ацикличен граф. Исто така, ќе го воведеме поимот за **граф од насоки (direction graph)** во ненасочениот транспортен систем. Овие теоретски резултати може да се користат за да се прилагодат алгоритмите за пресметување на множеството од сите минимални пат вектори за насочените транспортни системи да работат и за ненасочените транспортни системи.

Пред да ја изложиме теоријата со чија помош ќе ја најдеме врска помеѓу минимални пат вектори и протоците, мора да ги дадеме основните дефиниции и концепти прилагодени за ненасочени проточни транспортни системи. Овие дефиниции се слични со соодветните дефиниции за насочени графови кои се дадени во Глава 3 и подглава 4.1 но се прилагодени за ненасочени графови.

Двотерминален ненасочен проточен транспортен систем е ненасочен транспортен систем со два специфицирани јазли, извор s и сливник t ($s \neq t$), во којшто секој линк $\{u, v\} \in E$ има ненегативен капацитет $c(\{u, v\}) \geq 0$. И овде функцијата c се нарекува функција на капацитет. Повторно накратко ја користиме ознаката $G(V, E, c)$.

Проток во ненасочен транспортен систем $G(V, E, c)$ е функцијата $f: V \times V \rightarrow \mathbb{R}^+ \cup \{0\}$ што ги задоволува следниве три ограничувања:

1. **Ограничување на капацитетот:** $0 \leq f(u, v) \leq c(\{u, v\})$ за секој $(u, v) \in E$, т.е., протокот на линкот не може да го надмине капацитетот на линкот.
2. **Одржување на протокот:** За сите $v \in V$

$$f(V, v) - f(v, V) = \sum_{u \in V} f(u, v) - \sum_{w \in V} f(v, w) = \begin{cases} 0, & v \notin \{s, t\} \\ |f|, & v = s \\ -|f|, & v = t \end{cases}, \quad (4.5)$$

каде $|f|$ е вредноста на протокот. Со други зборови, вкупниот проток што влегува во јазолот v , $f(V, v)$, мора да биде еднаков на вкупниот проток што излегува од јазолот v , $f(v, V)$, $\forall v \in V \setminus \{s, t\}$; протокот што излегува од s и протокот што влегува во t е еднаков на вредноста на протокот.

3. **Правило за користење на линк во една насока:** За сите јазли $u, v \in V$, ако $f(u, v) > 0$, тогаш $f(v, u) = 0$. Со други зборови, секој проток користи даден линк само во една насока.

Дополнително претпоставуваме дека ако линкот $\{u, v\}$ не постои, т.е. $\{u, v\} \notin E$, тогаш $f(u, v) = 0$.

Како и кај насочените транспортни системи, и овде претпоставуваме дека линковите во ненасочениот транспортен систем се подредени т.е., $E = \{e_1, e_2, \dots, e_{|E|}\}$. Ако линковите ги гледаме како компоненти, транспортниот систем претставува повеќе - компонентен систем. Можеме да претпоставиме дека секоја компонента (линк e_i) може да работи со некое ниво $0 \leq x_i \leq c(e_i)$. Може да се забележи дека секој вектор од состојби дефинира капацитет на транспортниот систем, па ќе ја користиме ознаката $G(V, E, \vec{x})$ за двотерминален транспортен систем со извор s и сливник t во кој капацитетот на i -от линк e_i е еднаков на x_i . Ова означување, ни овозможува векторот од состојби да го претставуваме со ненасочен граф. За капацитетот на линкот $e_i = \{u, v\}$, ќе ја користиме ознаката x_{uv} наместо x_i .

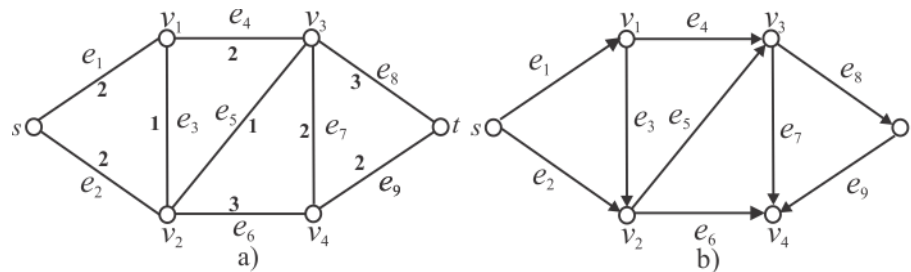
Сега ќе воведеме некои дефиниции кои ја дават врската помеѓу повеќе - состојбените системи и протокот во двотерминалните проточни транспортни системи.

Дефиниција 4.3 Нека $G(V, E, c)$ е двотерминален ненасочен проточен транспортен систем. За протокот f , дефинираме вектор од состојби \vec{x}_f индуциран од f со

$$x_i = |f(u, v) - f(v, u)| = \max(f(u, v), f(v, u)), \quad (4.6)$$

каде $e_i = \{u, v\}$.

Да забележиме дека равенката $|f(u, v) - f(v, u)| = \max(f(u, v), f(v, u))$ означува дека најмалку еден од протоците $f(u, v)$ и $f(v, u)$ е еднаков на 0.



Фигура 4.2 а) Ненасочен проточен транспортен систем; б) Еден граф од насоки за проточниот транспортен систем прикажан на а).

Бидејќи, протоците ги користат линковите во една од нивните две насоки, избираме едната насока да биде позитивна а другата негативна. На овој начин добиваме насочен граф $\hat{G}(V, \hat{E})$, кој ќе го наречеме граф од насоки за ненасочениот транспортен систем $G(V, E, c)$. Со користење на насоките што ги определивме со овој граф, ние дефинираме друг вектор сличен со векторот од состојби. (Да забележиме дека постојат повеќе можности за избор на ваков граф, и не е битно која можност ќе ја избереме). На Фигура 4.2 b) е прикажан еден можен избор на граф од насоки за ненасочениот проточен транспортен систем прикажан на Фигура 4.2 a).

Дефиниција 4.4 Нека $G(V, E, c)$ е двотерминален ненасочен проточен транспортен систем и $\hat{G}(V, \hat{E})$ е неговиот граф од насоки. За протокот f со вредност d , ние дефинираме **вектор индуциран од проток (flow vector)** \vec{x}'_f за **ниво d** индуциран од проток f и граф од насоки \hat{G} со

$$x'_i = \begin{cases} f(u, v), & (u, v) \in \hat{E} \text{ и } f(u, v) > 0 \\ -f(u, v), & (u, v) \in \hat{E} \text{ и } f(v, u) > 0 \\ 0, & f(u, v) = 0 \text{ и } f(v, u) = 0 \end{cases}, \quad (4.7)$$

каде $e_i = \{u, v\}$.

Забелешка 4.1 Векторот од состојби нема негативни координати, додека векторот индуциран од проток може да има негативни координати. Уште повеќе, за даден двотерминален ненасочен проточен транспортен систем (V, E, c) со граф од насоки $\hat{G}(V, \hat{E})$ и проток f , за секој i , $1 \leq i \leq |E|$, $x_i = |x'_i|$.

На пример, за транспортниот систем прикажан на Фигура 4.2, векторот $(0, 1, 1, 1, 0, 0, 0, 1, 0)$ е вектор од состојби индуциран од функцијата од проток f дефинирана со $f(e_i) = 1$, $i \in \{2, 3, 4, 8\}$, и $f(e_i) = 0$, $i \notin \{2, 3, 4, 8\}$. Со користење на насоките дефинирани со графот од насоки \hat{G} од Фигура 4.2 b), векторот $(0, 1, -1, 1, 0, 0, 0, 1, 0)$ е вектор индуциран од проток за ниво 1 индуциран од f и \hat{G} .

Понатаму се дадени дефиниции за пат вектор за ниво d и минимален пат вектор за ниво d за ненасочен транспортен систем.

Дефиниција 4.5 За даден ненасочен двотерминален проточен транспортен систем $G(V, E, c)$, векторот \vec{x} е пат вектор за ниво d , d -P, ако и само ако постои проток f со вредност $|f| = d$, така што за сите линкови $e_i = \{u, v\}$, $f(u, v) \leq x_i$. Со други зборови, \vec{x} е пат вектор за ниво d ако и само ако во двотерминалниот транспортен систем $G(V, E, \vec{x})$ проток d може да биде пренесен.

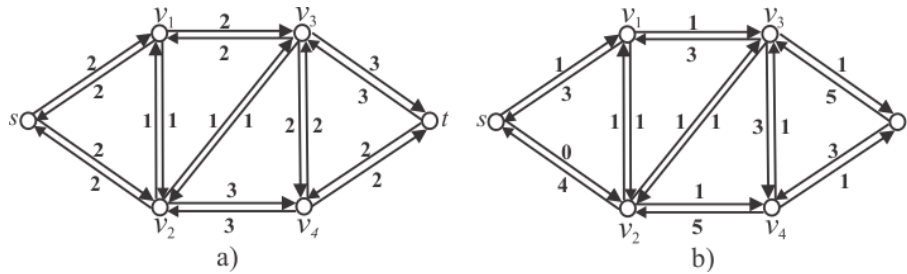
Дефиниција 4.6 Векторот од состојби \bar{x} е минимален пат вектор за ниво d , d -MP, ако и само ако двотерминалниот проточен транспортен систем $G(V, E, \bar{x})$ има максимален проток d , и за секој $\bar{y} \leq \bar{x}$, двотерминалниот проточен транспортен систем $G(V, E, \bar{y})$ има максимален проток помал од d .

Понатаму даваме некои познати факти, дадени во претходните подглави, но прилагодени за ненасочени транспортни системи.

Претпоставуваме дека имаме двотерминален проточен транспортен систем $G(V, E, c)$. Нека f е проток во G , и го разгледуваме парот од јазли $u, v \in V$. Дефинираме остаточен капацитет, $c_f(u, v)$, со

$$c_f(u, v) = \begin{cases} c(\{u, v\}) - f(u, v), & \{u, v\} \in E, \text{ и } f(u, v) \geq 0; \\ c(\{u, v\}) + f(u, v), & \{u, v\} \in E, \text{ и } f(v, u) \geq 0. \end{cases} \quad (4.8)$$

Ова значи, дека ако протокот f што е пуштен низ транспортниот систем, го користи линкот $\{u, v\}$ во насока од u до v , тогаш максималниот капацитет што може да биде пуштен низ линкот $\{u, v\}$ во насока од u до v е $c(\{u, v\}) - f(u, v)$ и максималниот проток што може да биде пуштен низ $\{u, v\}$ во насока од v до u е $c(\{u, v\}) + f(u, v)$.



Фигура 4.3 а) Остаточен транспортен систем индуциран од проток $f=0$; б) Остаточен транспортен систем индуциран од проток f претставен на Фигура 4.4 б).

Даден е проточниот транспортен систем $G(V, E)$ и проток f , остаточниот транспортен систем на G индуциран од f е $G_f(V, E_f)$, каде $E_f = \{(u, v) \in E \mid c_f(u, v) > 0\}$. Остаточниот транспортен систем секогаш е насочен, и за насочени и за ненасочени графови. На Фигура 4.3 а) и б) е прикажан остаточниот транспортен систем на транспортен систем прикажан на Фигура 4.2. На Фигура 4.3 а) е прикажан остаточен транспортен систем индуциран од проток $f=0$, додека на Фигура 4.3 б) е прикажан остаточен транспортен систем индуциран од протокот f прикажан на Фигура 4.4 б) ($|f|=3$ и $\bar{x}_f = (1, 2, 0, 1, 0, 2, 1, 2, 1)$).

Забелешка 4.2 За даден двотерминален ненасочен проточен транспортен систем $G(V, E, c)$, остаточниот транспортен систем индуциран од протокот за ниво $0, f$, каде $f(u, v) = 0$ за сите $\{u, v\} \in E$, е насочен транспортен систем $\tilde{G}(V, \tilde{E}, \tilde{c})$, каде $\tilde{E} = \{\{u, v\} | \{u, v\} \in E\}$ и $\tilde{c}(\{u, v\}) = c(\{u, v\})$, $\forall \{u, v\} \in E$.

Алгоритамот за наоѓање на проток (и максимален проток) е сличен на алгоритамот предложен од Ford-Fukerson (подглава 3.2), но сега се однесува за ненасочени графови. Всушност алгоритамот на Ford-Fukerson кога се применува на графот $\tilde{G}(V, \tilde{E}, \tilde{c})$, го дава протокот (и максималниот проток) за ненасочен транспортен систем [24]. За даден ненасочен проточен транспортен систем $G(V, E)$ и проток f , зголемувачкиот пат е дефиниран како прост пат p од s до t во остаточниот транспортен систем G_f .

Пресек (V_1, V_2) на ненасочен проточен транспортен систем $G(V, E, c)$ е поделба на множеството V на две подмножества V_1 и $V_2 = V \setminus V_1$, така што $s \in V_1$ и $t \in V_2$. Капацитетот на пресекот (V_1, V_2) е

$$c(V_1, V_2) = \sum_{u \in V_1} \sum_{v \in V_2} c(\{u, v\}). \quad (4.9)$$

Минимален пресек на мрежата е пресек чијшто капацитет е помал од капацитетите на сите други пресеци. Теоремата 3.1 (Теорема за максимален проток - минимален пресек) важи и за ненасочени транспортни системи. Уште повеќе, за даден проток f ,

$$f(V_1, V_2) = \sum_{u \in V_1} \sum_{v \in V_2} f(u, v) - \sum_{u \in V_1} \sum_{v \in V_2} f(v, u) = |f| \quad (4.10)$$

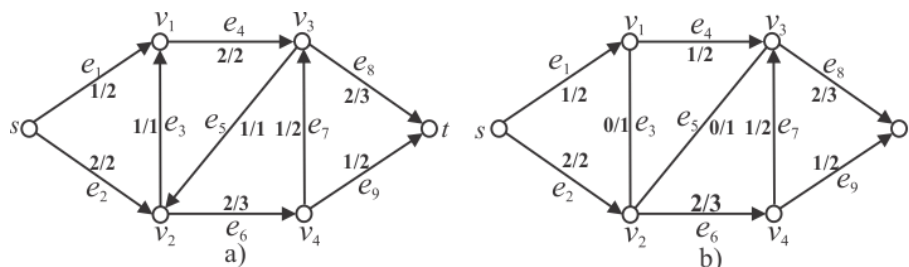
за било кој пресек (V_1, V_2) .

Во претходната подглава покажавме дека во насочен проточен транспортен систем, кој било вектор што одговара на проток со вредност d ако индуцира ацикличен граф е минимален пат вектор за ниво d . Овде ќе обезбедиме слични резултати за ненасочени транспортни системи.

Даден е проток f за $G(V, E, c)$, нека со E^f е означено множеството од сите линкови со позитивен проток т.е. $E^f = \{\{u, v\} | \{u, v\} \in E \text{ и } f(u, v) > 0\}$. Нетезинскиот граф $G(V, E^f)$ е граф индуциран од f .

Наредно што ќе докажеме се две теореми за ненасочени транспортни системи што се еквивалент на Теорема 4.1 од претходната подглава. Поточно, ќе покажеме дека протокот f со вредност d индуцира минимален пат вектор за ниво d ако и само E^f е ацикличен. Оваа е илустрирано на транспортните системи прикажани на Фигура 4.4. На Фигура 4.4 а) е прикажан проток со вредност 3, но векторот од состојби индуциран од него не е минимален пат вектор за ниво 3. Векторот од состојби индуциран од протокот со вредност 3 којшто е прикажан на Фигура 4.4 б) индуцира вектор од состојби што е помал од векторот прикажан на Фигура 4.4 а). Да забележиме дека графот индуциран од

протокот прикажан на Фигура 4.4 а) го содржи циклусот $\langle v_1, v_3, v_2, v_1 \rangle$, додека графот индуциран од протокот прикажан на Фигура 4.4 б) не содржи циклуси.



Фигура 4.4 а) Проток f со вредност 3 со $\vec{x}_f = (1, 2, 1, 2, 1, 2, 1, 2, 1)$, кој не е 3-MP; б) Проток со вредност 3 што индуцира 3-MP $(1, 2, 0, 1, 0, 2, 1, 2, 1)$ (ознаката a/b на секој од линковите означува дека капацитетот на соодветниот линк е b , и a е протокот што е искористен.).

Теорема 4.2 Нека \vec{x} е d -MP вектор во двотерминалниот ненасочен проточен транспортен систем $G(V, E, c)$ со извор s и сливник t . Тогаш, ненасочениот двотерминален проточен транспортен систем $G(V, E, \vec{x})$ има единствен проток со вредност d .

Доказ: Бидејќи \vec{x} е d -MP за $G(V, E, c)$, постои проток f во $G(V, E, \vec{x})$ со вредност d . Треба да покажеме дека f е единствен проток.

Прво да забележиме дека векторот од состојби \vec{x}_f индуциран од f е еднаков на \vec{x} , бидејќи во спротивно \vec{x}_f ќе биде помал d -P од \vec{x} , што е во контрадикција со претпоставката дека \vec{x} е d -MP. Затоа, за секој проток g во $G(V, E, \vec{x})$ со вредност d , ако $g(u, v) > 0$, тогаш $g(u, v) = x_{uv}$. Со други зборови, двата протоци може да се разликуваат само во насоката која тие ја користат за некој линк, т.е за секој проток g , или $g(u, v) = x_{uv}$ или $g(v, u) = x_{uv}$. За да покажеме дека постои само еден избор на насоки за линковите (избор на f), треба да покажеме дека графот $G(V, E^f)$ индуциран од f е ацикличен.

Да претпоставиме дека графот $G(V, E^f)$ има циклус \mathcal{C} . Земаме $m = \min\{f(u, v) \mid (u, v) \text{ е од циклусот } \mathcal{C}\}$. Функцијата $g : V \times V \rightarrow \mathbb{R}^+ \cup \{0\}$, дефинирана со

$$g(u, v) = \begin{cases} f(u, v) - m, & (u, v) \text{ е од циклусот } \mathcal{C} \\ 0, & \text{инаку} \end{cases}, \quad (4.11)$$

е исто така проток за ниво d , и векторот од состојби \vec{x}_g индуциран од g е помал од \vec{x} , што е повторно во контрадикција со нашата претпоставка дека \vec{x} е d -MP.

Сега, бидејќи $G(V, E^f)$ е ацикличен, постои тополошко сортирање на јазлите од V во однос на E^f . Нека (u, v) е произволен линк така што $x_{uv} > 0$. Без губење на општост, може да се претпостави дека во тополошкото сортирање u доаѓа пред v . Земаме V_1 да биде множеството од сите јазли од s до u , и $V_2 = V \setminus V_1$. Бидејќи f е проток и сите линкови во $G(V, E^f)$ поврзуваат јазол со помала ознака (вредност во однос на ова сортирање) со јазол со поголема ознака во однос на ова сортирање, протокот низ пресекот (V_1, V_2) е еднаков на:

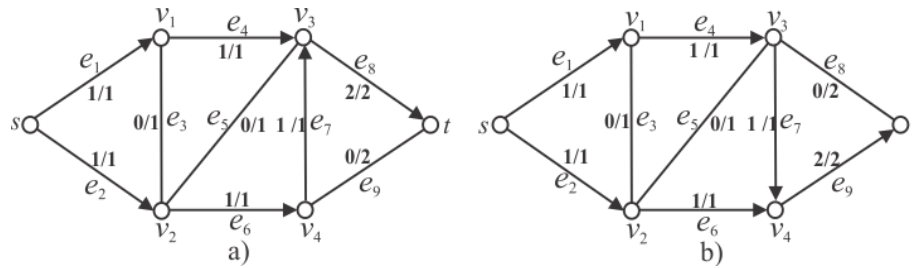
$$d = \sum_{w \in V_1} \sum_{w_1 \in V_2} f(w, w_1) - \sum_{w \in V_1} \sum_{w_1 \in V_2} f(w_1, w) = \sum_{w \in V_1} \sum_{w_1 \in V_2} f(w, w_1) = \sum_{w \in V_1} \sum_{w_1 \in V_2} x_{ww_1}. \quad (4.12)$$

Линкот $\{u, v\}$ лежи на пресекот (V_1, V_2) , и може да постои проток g што го користи овој линк во спротивна насока, па во овој случај пресекот низ (V_1, V_2) , ќе биде помал од d :

$$g(V_1, V_2) = \sum_{w \in V_1} \sum_{w_1 \in V_2} g(w, w_1) - \sum_{w \in V_1} \sum_{w_1 \in V_2} g(w_1, w) \leq \sum_{\substack{w \in V_1, w_1 \in V_2 \\ (w, w_1) \neq (u, v)}} x_{ww_1} - x_{uv} < d. \quad (4.13)$$

□

Бидејќи секој d -MP \vec{x} има единствен проток со вредност d , ќе го наречеме овој проток, **соодветен проток за d -MP, \vec{x}** . Да забележиме дека ова е случај само за d -MPs, но не и за секој d -P. На Фигура 4.5 се прикажани два протоци кои го користат линкот e_7 во две насоки за d -P $(1, 1, 0, 1, 0, 1, 1, 2, 2)$.



Фигура 4.5 Два протоци за d -P $(1, 1, 0, 1, 0, 1, 1, 2, 2)$, каде e_7 се користи во спротивни насоки.

Теорема 4.3 Нека f е проток што индуцира ацикличен граф $G(V, E^f)$ во двотерминалниот ненасочен проточен транспортен систем $G(V, E, c)$ со извор s и сливник t . Тогаш векторот од состојби \vec{x}_f е минимален пат вектор за ниво $|f|$, $|f|$ -MP.

Доказ: Јасно е дека \vec{x}_f е пат вектор за ниво d . Да претпоставиме дека постои помал пат вектор за ниво d од \vec{x}_f , да го означиме со \vec{y} . Тогаш постои проток g со $|g| = |f|$ што може да биде пренесен преку транспортниот систем $G(V, E, \vec{y})$. Без губење на општост, можеме да претпоставиме дека $(\vec{x}_f)_{uv} > y_{uv}$ и $f(u, v) > 0$ и $f(v, u) = 0$. Бидејќи $G(V, E^f)$ е ацикличен, се дефинира тополошко сортирање на јазлите од V , такашто u доаѓа пред v . Нека V_1 е множеството од јазли помеѓу s и u во ова сортирање, и $V_2 = V \setminus V_1$. Значи во однос на ова сортирање секој линк од $G(V, E^f)$ ги поврзува јазлите со помала ознака со јазли со поголема ознака, и поради тоа што f е проток, имаме дека $c(V_1, V_2)$ е сума од капацитетите на линковите што припаѓаат во овој пресек, т.е.

$$c(V_1, V_2) = |f| = \sum_{w \in V_1} \sum_{w_1 \in V_2} x_{ww_1}. \quad (4.14)$$

Од друга страна, бидејќи $(\vec{x}_f)_{uv} > y_{uv}$ имаме дека

$$|g| \leq \sum_{w \in V_1} \sum_{w_1 \in V_2} y_{ww_1} = \sum_{\substack{w \in V_1, w_1 \in V_2 \\ (w, w_1) \neq (u, v)}} y_{ww_1} - y_{uv} < \sum_{w \in V_1} \sum_{w_1 \in V_2} x_{ww_1} = |f|, \quad (4.15)$$

што е контрадикција со претпоставката дека $|g| = |f|$. \square

Од претходните две теореми заедно со дефиницијата на вектор индуциран од проток за ниво d , следува последицата.

Последица 4.1 Векторот од состојби \vec{x} е минимален пат вектор за ниво d во ненасочениот двотерминален проточен транспортен систем $G(V, E, c)$ со извор s , сливник t и граф од насоки $\hat{G}(V, \hat{E})$ ако и само ако постои единствен вектор индуциран од проток f за ниво d за кој графот $G(V, E^f)$ е ацикличен.

Според Последица 4.1, ако протокот f индуцира и минимален пат вектор за ниво d \vec{x} и вектор индуциран од проток \vec{x}' тогаш велиме дека \vec{x} е минимален пат вектор кој е соодветен на векторот индуциран од проток \vec{x}' , и обратно \vec{x}' е вектор индуциран од проток кој е соодветен за минималниот пат вектор. Ние ќе го наречени \vec{x}' **минимален вектор индуциран од проток** или **минимален d – MP проток**.

Од Забелешката 4.2 можеме да видиме дека остаточниот транспортен систем за даден ненасочен двотерминален проточен транспортен систем $G(V, E, c)$ индуциран од проток за ниво $0, f$, каде $f(u, v) = 0$ за сите $\{u, v\} \in E$, е графот $\tilde{G}(V, \tilde{E}, \tilde{c})$, каде $\tilde{E} = \{\{u, v\} \mid \{u, v\} \in E\}$ и $\tilde{c}(\{u, v\}) = c(\{u, v\})$, $\forall \{u, v\} \in E$. Јасно е дека секој зголемувачки пат p во $\tilde{G}(V, \tilde{E}, \tilde{c})$, дефинира проток со вредност 1. Уште попрецизно, функцијата $f: V \times V \rightarrow \mathbb{R}^+ \cup \{0\}$, дефинирана со

$$f(u, v) = \begin{cases} 1, & (u, v) \text{ е на патот } p, \\ 0, & \text{инаку} \end{cases}, \quad (4.16)$$

е проток со вредност 1. Овој проток го индуцира минималниот пат вектор за ниво 1 и минималниот вектор индуциран од проток за ниво 1, \vec{x}'_f .

Со следната теорема се определува дали векторот од состојби добиен како сума од прости патишта е минимален пат вектор за ниво d , d - MPs .

Теорема 4.4 За даден ненасочен двотерминален проточен транспортен систем $G(V, E, c)$ со извор s и сливник t , нека f_1, \dots, f_d се зголемувачки патишта во $\tilde{G}(V, \tilde{E}, \tilde{c})$ и $f = f_1 + \dots + f_d$. Ако $\forall (u, v) \in V \times V, f(u, v) \leq c(\{u, v\})$, тогаш \vec{x}_f е d -MP ако и само ако графот $G(V, E^f)$ е ацикличен.

Со помош на следната лема и теорема ќе докажеме дека сите протоци со некоја одредена вредност може да се добијат со сумирање на прости патишта. Ова е значајно за нас, бидејќи од наш интерес е да ги најдеме сите протоци за дадено ниво, а со алгоритмот на Ford-Fulkerson може да се најде само еден проток.

Лема 4.1 Нека f е функцијата од проток што индуцира минимален пат вектор за ниво d , во ненасочениот двотерминален проточен транспортен систем $G(V, E, c)$ со извор s , сливник t и граф од насоки $\hat{G}(V, \hat{E})$. Тогаш постои проток g што индуцира минимален пат вектор за ниво $d - 1$, и зголемувачки пат f_d на $\tilde{G}(V, \tilde{E}, \tilde{c})$ така што $f = g + f_d$. Уште повеќе, постои сортирање на јазлите во V кое може да биде истото тополошко сортирање за $G(V, E^g)$ и за $G(V, E^{f_d})$, како за $G(V, E^f)$.

Доказ: Нека f е проток така што \vec{x}_f е d -MP. Постои пат p во ацикличниот граф $G(V, E^f)$ што го поврзува s со t , кој исто така е зголемувачки пат во $\tilde{G}(V, \tilde{E}, \tilde{c})$. Дефинираме функција $f_d: V \times V \rightarrow \mathbb{R}^+ \cup \{0\}$ со

$$f_d(u, v) = \begin{cases} 1, & (u, v) \text{ е на патот } p \\ 0, & \text{инаку} \end{cases}, \quad (4.17)$$

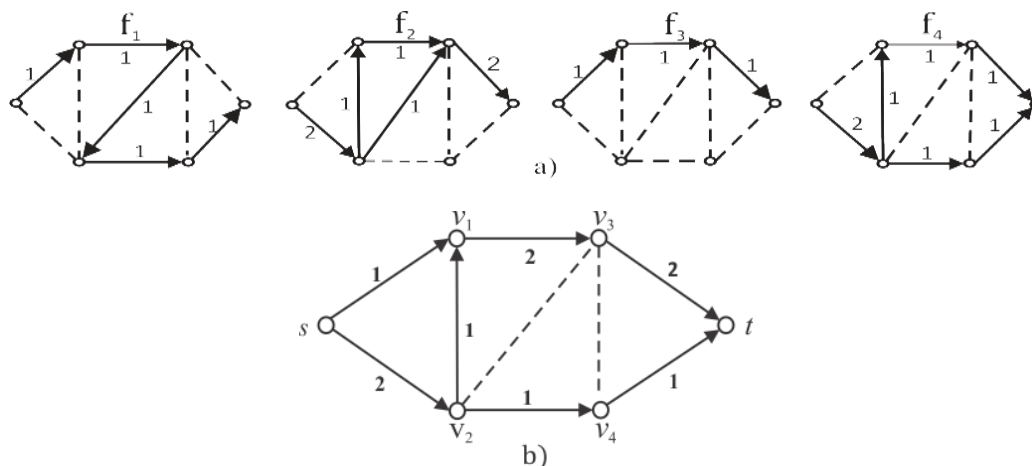
и $g: V \times V \rightarrow \mathbb{R}^+ \cup \{0\}$ со $g(u, v) = f(u, v) - f_d(u, v)$. Бидејќи за секој линк важи $g(u, v) \leq c(\{u, v\})$, и својството за одржување на проток е задоволено за сите јазли, функцијата g е проток со вредност $d - 1$.

Од друга страна, $G(V, E^g)$ е подграф од $G(V, E^f)$, па бидејќи тој е ацикличен и тополошкото сортирање на јазлите V што одговара на $G(V, E^f)$ може исто така да биде тополошко сортирање што одговара и на $G(V, E^g)$. Уште повеќе, бидејќи f_d беше добиено како пат во $G(V, E^f)$, истото сортирање може да биде тополошко сортирање и за $G(V, E^{f_d})$. Лемата е докажана со фактот дека $f = g + f_d$. \square

Следната теорема лесно се докажува со користење на индукција и Теорема 4.4.

Теорема 4.5 За даден ненасочен двотерминален проточен транспортен систем $G(V, E, c)$ со извор s и сливник t , и граф од насоки $\hat{G}(V, \hat{E})$, нека f е функцијата од проток така што \bar{x}_f е d -МП. Тогаш постојат зголемувачки патишта f_1, \dots, f_d во $\tilde{G}(V, \tilde{E}, \tilde{c})$ така што $f = f_1 + \dots + f_d$ и сите графови $G(V, E^{f_i})$, $i = \overline{1, d}$, може да имаат исто тополошко сортирање.

Да претпоставиме дека некои функции што индуцираат d -МПs може да бидат добиени како сума од 1-МПs со различни тополошки сортирања, но од последната теорема следува дека секогаш постои начин протокот да се претстави како сума од 1-МПs. На пример, протокот со вредност 3 што индуцира 3-МП прикажан на Фигура 4.6 b) може да се добие од протоците f_1 и f_2 , кои индуцираат графови што немаат исто тополошко сортирање. ($G(V, E^{f_1})$ нема исто тополошко сортирање со графот прикажан на Фигура 4.6 b)) Меѓутоа, истиот проток може да биде добиен и од протоците f_3 и f_4 .



Фигура 4.6 а) f_1 и f_3 се протоци со вредност 1, додека f_2 и f_4 се протоци со вредност 2 за проточниот транспортен систем во Фигура 4.2.а); б) Проток со вредност 3 што може да биде добиен од протоците f_1 и f_2 или од протоците f_3 и f_4 .

Резултатите од овој дел може да се искористат, за модифицирање на алгоритмите за наоѓање на d МПs во насочени транспортни системи, за ненасочени транспортни системи.

Глава 5

5. Алгоритми за пресметување на минимални пат вектори за ниво d базирани на својството на ацикличност на графови индуцирани од проток

Во оваа глава ќе предложиме алгоритми за пресметување на минимални пат вектори за ниво d . Во Глава 4, со Теорема 4.1 за насочени транспортни системи и со Теорема 4.2 и Теорема 4.3 за ненасочени транспортни системи докажавме дека графот што е индуциран од минимален пат вектор е ацикличен. Па токму на оваа идеја се базираат сите алгоритми што ќе ги предложиме во оваа глава. Поточно, кога ќе добиваме кандидат за минимален пат вектор ќе вршиме проверка дали графот што го индуцира кандидатот за минимален пат вектор е ацикличен или има циклус. Најпрво ќе ги разгледаме основните верзии на алгоритмите за насочени и за ненасочени графови. Во овие алгоритми прво се добиваат сите кандидати за минимални пат вектори. Тие може да се добијат со кој било од познатите алгоритми за наоѓање на минимални пат вектори за ниво d . Ние овде го земаме алгоритмот предложен во [51] за наоѓање на минималните пат вектори. Откако ќе се најдат сите минимални пат вектори со примена на Теорема 4.1 за насочените графови, а со примена на Теорема 4.2 и Теорема 4.3 за ненасочените графови, за секој кандидат за минимален пат вектор се конструира графот $G(V, E^{l_{\vec{x}}})$, каде $l_{\vec{x}}$ е псевдопроток. Ако во овој граф не постои циклус, во тој случај кандидатот за минимален пат вектор е минимален пат вектор, во спротивно се брише од множеството на кандидати за минимални пат вектори. Да го дадеме прво псевдокодот на алгоритмот за насочени графови.

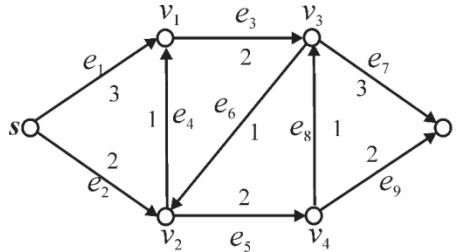
Алгоритам 5.1 – Алгоритам за наоѓање на минимални пат вектори за ниво d за насочени транспортни системи $(G(V, E, c))$

Чекор 1. Со користење на алгоритмот во [51], да се најде множеството Q од сите протоци од кои се индуцираат кандидатите за минимални пат вектори за ниво d , d -MP.

Чекор 2. За секој кандидат за минимален пат вектор \vec{y} се проверува дали постои циклус во $G(V, E^{l_{\vec{y}}})$, ако постои циклус се брише кандидатот за минимален пат вектор за ниво d од Q .

Проверката за постоење на циклуси во графот може на едноставен начин да се направи со користење на алгоритмот за пребарување по длабочина - DFS [3]. Овој алгоритам има временска сложеност $O(|E|)$. Како резултат на оваа временската сложеност на предложениот алгоритам е $O(|E| \lambda)$, каде λ е горната граница на бројот на добиени кандидати со алгоритмот предложен во [51]. Сложеноста на овој алгоритам е значајно помала од сложеноста на алгоритмот предложен во [34], $O(|V|^2 |E|^{3/2} \lambda)$.

Сега да го илустрираме Алгоритмот 5.1 на транспортниот систем претставен на Фигура 5.1.



Фигура 5.1 Насочен транспортен систем.

Како што видовме Алгоритам 5.1 се базира на алгоритмот во [51], т.е. кандидатите за минимални пат вектори се наоѓаат со алгоритмот на Lin, Jane и Yuan. Се разликуваат само во проверката дали некој кандидат е минимален пат вектор. Со алгоритмот предложен во [51] проверката дали кандидатите за минимални пат вектори се минимални пат вектори се врши со заемна споредба помеѓу кандидатите, додека во предложениот алгоритам се проверува дали постои циклус во графот $G(V, E^{l_{\vec{x}}})$.

Пример 5.1. Го разгледуваме транспортниот систем прикажан на Фигура 5.1 која има 9 линкови и 6 јазли. Векторот од минимални капацитети е $(3, 2, 2, 1, 2, 1, 3, 1, 2)$. Минималните патишта се $\mathcal{P}_1 = \{e_1, e_3, e_6, e_5, e_9\}$, $\mathcal{P}_2 = \{e_1, e_3, e_7\}$, $\mathcal{P}_3 = \{e_2, e_4, e_3, e_7\}$, $\mathcal{P}_4 = \{e_2, e_5, e_8, e_7\}$ и $\mathcal{P}_5 = \{e_2, e_5, e_9\}$. Се бара ниво на работа на системот $d = 3$, фамилијата од минималните пат вектори за ниво 3, 3 -MPs, е добиена на следниот начин:

Чекор 1. 1 -MP = $\{(1, 0, 1, 0, 1, 1, 0, 0, 1), (1, 0, 1, 0, 0, 0, 1, 0, 0), (0, 1, 1, 1, 0, 0, 1, 0, 0), (0, 1, 0, 0, 1, 0, 1, 1, 0), (0, 1, 0, 0, 1, 0, 0, 0, 1)\}$,

$$c_{\vec{x}_1} = \min\{3, 2, 1, 2, 2\} = 1, \quad c_{\vec{x}_2} = \min\{3, 2, 3\} = 2, \quad c_{\vec{x}_3} = \min\{2, 1, 2, 3\} = 1,$$

$$c_{\vec{x}_4} = \min\{2, 2, 1, 3\} = 1 \text{ и } c_{\vec{x}_5} = \min\{2, 2, 2\} = 2.$$

Се наоѓаат сите вектори $c_{\vec{f}} = (c_{f_1}, c_{f_2}, c_{f_3}, c_{f_4}, c_{f_5})$ кои ги задоволуваат ограничувањата (A.1) - (A.3):

$$\begin{aligned} c_{f_1} + c_{f_2} + c_{f_3} + c_{f_4} + c_{f_5} &= 3, \\ (c_{f_1}, c_{f_2}, c_{f_3}, c_{f_4}, c_{f_5}) &\leq (1, 2, 1, 1, 2), \\ \sum_j \{c_{f_j} \mid x_{j_1} = 1\} &= c_{f_1} + c_{f_2} \leq 3, \\ \sum_j \{c_{f_j} \mid x_{j_2} = 1\} &= c_{f_3} + c_{f_4} + c_{f_5} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_3} = 1\} &= c_{f_1} + c_{f_2} + c_{f_3} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_4} = 1\} &= c_{f_3} \leq 1, \\ \sum_j \{c_{f_j} \mid x_{j_5} = 1\} &= c_{f_1} + c_{f_4} + c_{f_5} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_6} = 1\} &= c_{f_1} \leq 1, \\ \sum_j \{c_{f_j} \mid x_{j_7} = 1\} &= c_{f_2} + c_{f_3} + c_{f_4} \leq 3, \\ \sum_j \{c_{f_j} \mid x_{j_8} = 1\} &= c_{f_4} \leq 1, \\ \sum_j \{c_{f_j} \mid x_{j_9} = 1\} &= c_{f_1} + c_{f_5} \leq 1. \end{aligned}$$

Добиени се следниве седум вектори $c_{\vec{f}_1} = (0, 1, 0, 1, 1)$, $c_{\vec{f}_2} = (0, 1, 1, 0, 1)$,

$$c_{\vec{f}_3} = (0, 1, 1, 1, 0), \quad c_{\vec{f}_4} = (1, 0, 1, 1, 0), \quad c_{\vec{f}_5} = (1, 1, 0, 1, 0), \quad c_{\vec{f}_6} = (0, 2, 0, 0, 1),$$

$$c_{\vec{f}_7} = (0, 2, 0, 1, 0).$$

Со помош на $c_{\vec{f}_j}$, $j = 1, 2, \dots, 7$, се добиваат 3 – *MP* кандидатите

$$\vec{y} = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9).$$

$$\begin{aligned} \vec{y}_1 &= \vec{x}_1 + \vec{x}_4 + \vec{x}_5 = (1, 2, 1, 0, 2, 0, 2, 1, 1), \\ \vec{y}_2 &= \vec{x}_2 + \vec{x}_3 + \vec{x}_5 = (1, 2, 2, 1, 1, 0, 2, 0, 1), \\ \vec{y}_3 &= \vec{x}_2 + \vec{x}_3 + \vec{x}_4 = (1, 2, 2, 1, 1, 0, 3, 1, 0), \\ \vec{y}_4 &= \vec{x}_1 + \vec{x}_3 + \vec{x}_4 = (1, 2, 2, 1, 2, 1, 2, 1, 1), \\ \vec{y}_5 &= \vec{x}_1 + \vec{x}_2 + \vec{x}_4 = (2, 1, 2, 0, 2, 1, 2, 1, 1), \\ \vec{y}_6 &= 2\vec{x}_2 + \vec{x}_5 = (2, 1, 2, 0, 1, 0, 2, 0, 1), \end{aligned}$$

$$\vec{y}_7 = 2\vec{x}_2 + \vec{x}_4 = (2, 1, 2, 0, 1, 0, 3, 1, 0).$$

Чекор 2. Графовите што се индуцирани од векторите $\vec{y}_1, \vec{y}_2, \vec{y}_3, \vec{y}_6$ и \vec{y}_7 не содржат циклуси. Додека графовите што се индуцирани од векторите \vec{y}_4 и \vec{y}_5 содржат циклуси. Во графот индуциран од \vec{y}_4 тоа се циклусите $\langle v_2, v_1, v_3, v_2 \rangle$ и $\langle v_2, v_4, v_3, v_2 \rangle$, додека во графот индуциран од \vec{y}_5 тоа е циклусот $\langle v_2, v_4, v_3, v_2 \rangle$.

Множеството од минимални пат вектори за ниво e

$$3-MP = \{(1, 2, 1, 0, 2, 0, 2, 1, 1), (1, 2, 2, 1, 1, 0, 2, 0, 1), (1, 2, 2, 1, 1, 0, 3, 1, 0), (2, 1, 2, 0, 1, 0, 2, 0, 1), (2, 1, 2, 0, 1, 0, 3, 1, 0)\}.$$

Во овој пример, 7 кандидати за минимални пат вектори се добиени: 5 се минимални пат вектори и 2 не се минимални пат вектори.

Теорема 4.2 и Теорема 4.3 може да се искористат за модифицирање на алгоритмот предложен во [51], за да може модифицираниот алгоритам да се применува и на ненасочени транспортни системи. Всушност, наместо алгоритмот да се применува на оригиналниот транспортен систем $G(V, E, c)$, ќе се применува на транспортен систем $\tilde{G}(V, \tilde{E}, \tilde{c})$. Проверката дали одреден кандидат за $d-MP$ е $d-MP$ се врши со проверување дали постои циклус во графот $G(V, E^f)$ за проток f што индуцира кандидат за минимален пат вектор. Чекорите од алгоритмот за ненасочени транспортни системи се дадени подолу.

Алгоритам 5.2 – Алгоритам за наоѓање на минимални пат вектори за ниво d на ненасочен транспортен систем $(G(V, E, c))$

Чекор 1. Се конструира графот $\tilde{G}(V, \tilde{E}, \tilde{c})$.

Чекор 2. Се применува алгоритмот предложен во [51] на графот $\tilde{G}(V, \tilde{E}, \tilde{c})$ за да се најде множеството Q од сите протоци што индуцираат вектори кои се кандидати за $d-MP$.

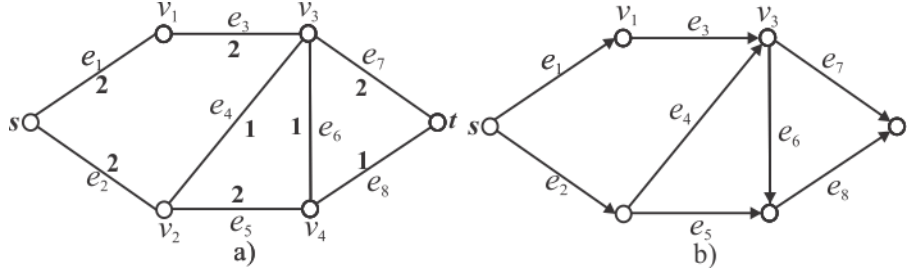
Чекор 3. За секој кандидат f се проверува дали постои циклус во графот $G(V, E^f)$ и ако постои кандидатот се брише од Q .

Чекор 4. Се наоѓаат сите $d-MPs$ за протоците во Q .

Како што кажавме и претходно кога го дадовме алгоритмот за насочени транспортни системи, проверката дали постои циклус во графот може да биде направена со користење на DFS. Поради тоа временската комплексност на овој алгоритам е $O(|E|\lambda)$, каде λ е горната граница за бројот на кандидати што се добиваат со алгоритмот на Lin, Janе и Yuan. Ова е истата комплексност како и во случај кога разгледувавме насочен транспортен систем (Алгоритам 5.1). Оваа комплексност е значително помала од комплексноста на алгоритмот даден во [34], која е еднаква на $O(|V|^2|E|^{3/2}\lambda)$. Да забележиме дека некои $d-MPs$ ќе бидат добиени повеќе пати, па тие треба да се игнорираат. Со користење на бинарно пребарување, комплексноста ќе биде $\log(|Q|)$.

Бидејќи Q има најмалку $m^{|E|}$ елементи, каде m е максималниот капацитет на линковите, временската комплексност на овој дел е еднаква на $|E|\log(m)$. Па комплексноста на алгоритмот ќе биде $O(|E|\lambda + |E|\log(m))$.

Пример 5.2 Да го разгледаме транспортниот прикажана на Фигура 5.2



Фигура 5.2 а) Ненасочен транспортен систем; б) граф од насоки за ненасочениот транспортен

Прво да го примениме алгоритмот предложен во [51] на транспортниот систем од Фигура 5.2.

$d = 2$:

Минимални патишта се $\mathcal{P}_1 = \{e_1, e_3, e_4, e_5, e_8\}$ (e_4 се користи во спротивна насока), $\mathcal{P}_2 = \{e_1, e_3, e_6, e_8\}$, $\mathcal{P}_3 = \{e_1, e_3, e_7\}$, $\mathcal{P}_4 = \{e_2, e_4, e_6, e_8\}$, $\mathcal{P}_5 = \{e_2, e_4, e_7\}$, $\mathcal{P}_6 = \{e_2, e_5, e_6, e_7\}$ (e_6 се користи во спротивна насока) и $\mathcal{P}_7 = \{e_2, e_5, e_8\}$.

1-MP = $\{(1, 0, 1, -1, 1, 0, 0, 1), (1, 0, 1, 0, 0, 1, 0, 1), (1, 0, 1, 0, 0, 0, 1, 0), (0, 1, 0, 1, 0, 1, 0, 1), (0, 1, 0, 1, 0, 0, 1, 0), (0, 1, 0, 0, 1, -1, 1, 0), (0, 1, 0, 0, 1, 0, 0, 1)\}$

За дадено ниво $d = 2$, фамилијата од сите 2-MPs е добиена на следниот начин:

$c_{\bar{x}_1} = \min\{2, 2, 1, 2, 1\} = 1$, $c_{\bar{x}_2} = \min\{2, 2, 1, 1\} = 1$, $c_{\bar{x}_3} = \min\{2, 2, 2\} = 2$, $c_{\bar{x}_4} = \min\{2, 1, 1\} = 1$, $c_{\bar{x}_5} = \min\{2, 1, 2\} = 1$, $c_{\bar{x}_6} = \min\{2, 2, 1, 2\} = 1$ и $c_{\bar{x}_7} = \min\{2, 2, 1\} = 1$

Да ги најдеме сите вектори $c_{\bar{f}} = (c_{f_1}, c_{f_2}, c_{f_3}, c_{f_4}, c_{f_5}, c_{f_6}, c_{f_7})$ што ги задоволуваат ограничувањата (A.1) - (A.3) :

$$\begin{aligned} c_{f_1} + c_{f_2} + c_{f_3} + c_{f_4} + c_{f_5} + c_{f_6} + c_{f_7} &= 2, \\ (c_{f_1}, c_{f_2}, c_{f_3}, c_{f_4}, c_{f_5}, c_{f_6}, c_{f_7}) &\leq (1, 1, 2, 1, 1, 1, 1), \\ \sum_j \{c_{f_j} \mid x_{j_1} = 1\} &= c_{f_1} + c_{f_2} + c_{f_3} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_2} = 1\} &= c_{f_4} + c_{f_5} + c_{f_6} + c_{f_7} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_3} = 1\} &= c_{f_1} + c_{f_2} + c_{f_3} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_4} = 1\} &= -c_{f_1} + c_{f_4} + c_{f_5} \leq 1, \end{aligned}$$

$$\begin{aligned}\sum_j \{c_{f_j} \mid x_{j_5} = 1\} &= c_{f_1} + c_{f_4} + c_{f_5} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_6} = 1\} &= c_{f_2} + c_{f_4} - c_{f_6} \leq 1, \\ \sum_j \{c_{f_j} \mid x_{j_7} = 1\} &= c_{f_3} + c_{f_5} + c_{f_6} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_8} = 1\} &= c_{f_1} + c_{f_2} + c_{f_4} + c_{f_7} \leq 1.\end{aligned}$$

Добиваме 14 вектори: $c_{\bar{f}_1} = (0, 0, 0, 0, 0, 1, 1)$, $c_{\bar{f}_2} = (0, 0, 0, 0, 1, 0, 1)$, $c_{\bar{f}_3} = (0, 0, 0, 0, 1, 1, 0)$, $c_{\bar{f}_4} = (0, 0, 0, 1, 0, 1, 0)$, $c_{\bar{f}_5} = (0, 0, 1, 0, 0, 1, 0)$, $c_{\bar{f}_6} = (0, 0, 1, 0, 1, 0, 0)$, $c_{\bar{f}_7} = (0, 0, 1, 1, 0, 0, 0)$, $c_{\bar{f}_8} = (0, 1, 0, 0, 0, 1, 0)$, $c_{\bar{f}_9} = (0, 1, 0, 0, 1, 0, 0)$, $c_{\bar{f}_{10}} = (0, 1, 1, 0, 0, 0, 0)$, $c_{\bar{f}_{11}} = (1, 0, 0, 0, 0, 1, 0)$, $c_{\bar{f}_{12}} = (1, 0, 0, 0, 1, 0, 0)$, $c_{\bar{f}_{13}} = (1, 0, 1, 0, 0, 0, 0)$ и $c_{\bar{f}_{14}} = (0, 0, 2, 0, 0, 0, 0)$.

Со помош на $c_{\bar{f}_j}$ се добиваат 2- MP кандидатите $\bar{y} = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$.

$$\begin{aligned}\bar{y}_1 &= \bar{x}_6 + \bar{x}_7 = (0, 2, 0, 0, 2, -1, 1, 1), \\ \bar{y}_2 &= \bar{x}_5 + \bar{x}_7 = (0, 2, 0, 1, 1, 0, 1, 1), \\ \bar{y}_3 &= \bar{x}_5 + \bar{x}_6 = (0, 2, 0, 1, 1, -1, 2, 0), \\ \bar{y}_4 &= \bar{x}_4 + \bar{x}_6 = (0, 2, 0, 1, 1, 0, 1, 1), \\ \bar{y}_5 &= \bar{x}_3 + \bar{x}_6 = (1, 1, 1, 0, 1, -1, 2, 0), \\ \bar{y}_6 &= \bar{x}_3 + \bar{x}_5 = (1, 1, 1, 1, 0, 0, 2, 0), \\ \bar{y}_7 &= \bar{x}_3 + \bar{x}_4 = (1, 1, 1, 1, 0, 1, 1, 1), \\ \bar{y}_8 &= \bar{x}_2 + \bar{x}_6 = (1, 1, 1, 0, 1, 0, 1, 1), \\ \bar{y}_9 &= \bar{x}_2 + \bar{x}_5 = (1, 1, 1, 1, 0, 1, 1, 1), \\ \bar{y}_{10} &= \bar{x}_2 + \bar{x}_3 = (2, 0, 2, 0, 0, 1, 1, 1), \\ \bar{y}_{11} &= \bar{x}_1 + \bar{x}_6 = (1, 1, 1, -1, 2, -1, 1, 1), \\ \bar{y}_{12} &= \bar{x}_1 + \bar{x}_5 = (1, 1, 1, 0, 1, 0, 1, 1), \\ \bar{y}_{13} &= \bar{x}_1 + \bar{x}_3 = (2, 0, 2, -1, 1, 0, 1, 1), \\ \bar{y}_{14} &= 2\bar{x}_3 = (2, 0, 2, 0, 0, 0, 2, 0).\end{aligned}$$

Кандидати што се повторуваат или не се минимални пат вектори се:

$$\begin{aligned}\bar{y}_4 &= \bar{y}_2, \\ \bar{y}_9 &= \bar{y}_7, \\ \bar{y}_{11} &> \bar{y}_7 \Rightarrow \bar{y}_{11} \text{ не е } 2\text{-}MP,\end{aligned}$$

$$\vec{y}_{12} = \vec{y}_8.$$

$$Q_2 = \{(0, 2, 0, 0, 2, 1, 1, 1), (0, 2, 0, 1, 1, 0, 1, 1), (0, 2, 0, 1, 1, 1, 2, 0), (1, 1, 1, 0, 1, 1, 2, 0), (1, 1, 1, 1, 0, 0, 2, 0), (1, 1, 1, 1, 0, 1, 1, 1), (1, 1, 1, 0, 1, 0, 1, 1), (2, 0, 2, 0, 0, 1, 1, 1), (2, 0, 2, 1, 1, 0, 1, 1), (2, 0, 2, 0, 0, 0, 2, 0)\}$$

$d = 3$:

За $d = 3$ се разликува само условот (A.1) т.е. во овој случај

$$c_{f_1} + c_{f_2} + c_{f_3} + c_{f_4} + c_{f_5} + c_{f_6} + c_{f_7} = 3.$$

Добиваме десет вектори $c_{\vec{f}_1} = (1, 0, 1, 0, 1, 0, 0)$, $c_{\vec{f}_2} = (1, 0, 1, 0, 0, 1, 0)$, $c_{\vec{f}_3} = (1, 0, 0, 0, 1, 1, 0)$, $c_{\vec{f}_4} = (0, 1, 1, 0, 1, 0, 0)$, $c_{\vec{f}_5} = (0, 1, 0, 0, 1, 1, 0)$, $c_{\vec{f}_6} = (0, 0, 1, 0, 1, 0, 1)$, $c_{\vec{f}_7} = (0, 0, 1, 0, 0, 1, 1)$, $c_{\vec{f}_8} = (0, 0, 1, 1, 0, 1, 0)$, $c_{\vec{f}_9} = (0, 0, 2, 1, 0, 0, 0)$ и $c_{\vec{f}_{10}} = (0, 0, 2, 0, 0, 0, 1)$.

Со помош на $c_{\vec{f}_j}$ се добиваат 3- MP кандидатите $\vec{y} = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$.

$$\begin{aligned}\vec{y}_1 &= \vec{x}_1 + \vec{x}_3 + \vec{x}_5 = (2, 1, 2, 0, 1, 0, 2, 1), \\ \vec{y}_2 &= \vec{x}_1 + \vec{x}_3 + \vec{x}_6 = (2, 1, 2, -1, 2, -1, 2, 1), \\ \vec{y}_3 &= \vec{x}_1 + \vec{x}_5 + \vec{x}_6 = (1, 2, 1, 0, 2, -1, 2, 1), \\ \vec{y}_4 &= \vec{x}_2 + \vec{x}_3 + \vec{x}_5 = (2, 1, 2, 1, 0, 1, 2, 1), \\ \vec{y}_5 &= \vec{x}_2 + \vec{x}_5 + \vec{x}_6 = (1, 2, 1, 1, 1, 0, 2, 1), \\ \vec{y}_6 &= \vec{x}_3 + \vec{x}_5 + \vec{x}_7 = (1, 2, 1, 1, 1, 0, 2, 1), \\ \vec{y}_7 &= \vec{x}_3 + \vec{x}_6 + \vec{x}_7 = (1, 2, 1, 0, 2, -1, 2, 1), \\ \vec{y}_8 &= \vec{x}_3 + \vec{x}_4 + \vec{x}_6 = (1, 2, 1, 1, 1, 0, 2, 1), \\ \vec{y}_9 &= 2\vec{x}_3 + \vec{x}_4 = (2, 1, 2, 1, 0, 1, 2, 1), \\ \vec{y}_{10} &= 2\vec{x}_3 + \vec{x}_7 = (2, 1, 2, 0, 1, 0, 2, 1).\end{aligned}$$

Кандидати што се повторуваат или не се минимални пат вектори се:

$$\begin{aligned}\vec{y}_2 &> \vec{y}_1 \Rightarrow \vec{y}_2 \text{ не е } 3\text{-}MP \\ \vec{y}_6 &= \vec{y}_5, \\ \vec{y}_7 &= \vec{y}_3, \\ \vec{y}_8 &= \vec{y}_5, \\ \vec{y}_9 &= \vec{y}_4, \\ \vec{y}_{10} &= \vec{y}_1. \\ Q_3 &= \{(2, 1, 2, 0, 1, 0, 2, 1), (1, 2, 1, 0, 2, -1, 2, 1), (2, 1, 2, 1, 0, 1, 2, 1), (1, 2, 1, 1, 1, 0, 2, 1)\}\end{aligned}$$

Со примена на Алгоритмот 5.2

$$Q'_2 = \{(0, 2, 0, 0, 2, 1, 1, 1), (0, 2, 0, 1, 1, 0, 1, 1), (0, 2, 0, 1, 1, -1, 2, 0), (0, 2, 0, 1, 1, 0, 1, 1), (1, 1, 1, 0, 1, -1, 2, 0), (1, 1, 1, 1, 0, 0, 2, 0), (1, 1, 1, 1, 0, 1, 1, 1), (1, 1, 1, 0, 1, 0, 1, 1), (1, 1, 1, 1, 0, 1, 1, 1), (2, 0, 2, 0, 0, 1, 1, 1), (1, 1, 1, -1, 2, -1, 1), (1, 1, 1, 0, 1, 0, 1, 1), (2, 0, 2, -1, 1, 0, 1, 1), (2, 0, 2, 0, 0, 0, 2, 0)\}.$$

Графот индуциран од векторот $(1, 1, 1, -1, 2, -1, 1, 1)$ има циклус. Тоа е циклусот $\langle v_3, v_2, v_4, v_3 \rangle$. Од каде следува дека $(1, 1, 1, -1, 2, -1, 1, 1)$ не е минимален пат вектор за ниво 2.

$$Q'_3 = \{(2, 1, 2, 0, 1, 0, 2, 1), (2, 1, 2, -1, 2, -1, 2, 1), (1, 2, 1, 0, 2, -1, 2, 1), (2, 1, 2, 1, 0, 1, 2, 1), (1, 2, 1, 1, 1, 0, 2, 1), (1, 2, 1, 1, 1, 0, 2, 1), (1, 2, 1, 0, 2, -1, 2, 1), (1, 2, 1, 1, 1, 0, 2, 1), (2, 1, 2, 1, 0, 1, 2, 1), (2, 1, 2, 0, 1, 0, 2, 1)\}.$$

Графот индуциран од векторот $(2, 1, 2, -1, 2, -1, 2, 1)$ има циклус. Тоа е циклусот $\langle v_3, v_2, v_4, v_3 \rangle$. Од каде следува дека $(2, 1, 2, -1, 2, -1, 2, 1)$ не е минимален пат вектор за ниво 3.

Во продолжение на оваа глава ќе предложиме други алгоритми за наоѓање на минимални пат вектори за ниво d . Сите тие алгоритми се базираат на својството на ацикличност на графови.

Во првата подглава е предложен алгоритам за пресметување на минимални пат вектори за ниво кое работи и за насочени и за ненасочени транспортни системи. Користејќи го фактот дека минималните пат вектори индуцираат ацикличен граф, се дефинира релација на достапност во ацикличниот граф која дефинира подредување на јазлите во графот. Исто така, се наоѓа и матрицата на достапност. Врз основа на оваа матрица забележуваме дали графот што го формира некој минимален пат вектор е ацикличен или содржи циклус. Доколку на дијагоналата во оваа матрица сите елементи се еднакви на нула, тогаш соодветниот граф е ацикличен, па и векторот од состојби што го индуцира овој граф ќе биде минимален пат вектор. Во општ случај овој алгоритам има експоненцијална сложеност. Но во одредени случаи за единица време може да се види дали постои циклус во графот што е индуциран од кандидат за минимален пат вектор кој се добива како збир од два минимални пат вектори. За минималните пат вектори се наоѓаат матриците на достапност. Доколку во една од матриците има единица на позицијата (i, j) а во другата матрица има единица на позицијата (j, i) , тогаш веднаш го елиминираме овој кандидат.

Во втората подглава се предложени алгоритми за наоѓање на минимални пат вектори во насочен двотерминален повеќе - состојбен проточен транспортен систем. Прво е предложен алгоритам што го користи алгоритмот на Ford-Fulkerson. Но со овој алгоритам ние можеме да најдеме еден проток за ниво d . Со користење на Теорема 4.1 можеме да најдеме еден минимален пат вектор за ниво d . Затоа овде ќе ја дадеме и врската помеѓу минималните пат вектори и врз основа на таа врска ќе ги наоѓаме останатите минимални пат вектори. Во продолжение од подглавата ќе предложиме алгоритам кој ги

минимални пат вектори. Во продолжение од подглавата ќе предложиме алгоритам кој ги наоѓа минималните пат вектори за максимално ниво. Идеја на овој алгоритам произлегува поради тоа што остаточниот транспортен систем што се добива со користење на алгоритмот на Ford-Fulkerson за максимално ниво може да биде поделен во строго сврзани компоненти. Овие компоненти се користат за добивање на минималните пресеци. Секој минимален пат вектор за максималното ниво може да се добие со спојување на подвекторите за соодветните строго поврзани компоненти со линковите кои лежат на минималните пресеци.

Во третата подглава е предложен подобрен алгоритам за наоѓање на минимални пат вектори во ненасочен двотерминален повеќе - состојбен проточен транспортен систем. Подобрувањето се состои со тоа што воведуваме ограничувања кои вектори да ги собираме. На тој начин се намалува бројот на кандидати за минимални пат вектори. Всушност, ќе покажеме дека секој минимален пат вектор што се добива со собирање на вектори кои имаат соодветни координати со спротивни знаци може да се добие и на друг начин, без користење на вакви вектори. Ова го користиме за подобрување на Алгоритам 5.2 со тоа што нема да ги земаме предвид векторите кои имаат спротивни знаци на соодветните координати.

5.1 Алгоритам со контрола на циклуси

Со најголемиот број на алгоритми што ги разгледавме за наоѓање на минимални пат вектори, се добиваат и кандидати за минимални пат вектори кои не се минимални. Овие кандидати за минимални пат вектори што не се минимални се елиминираат на тој начин што се споредуваат заемно, поточно, ако даден кандидат за минимален пат вектор е поголем од друг кандидат вектор тогаш тој не е минимален и се елиминира. Меѓутоа оваа процедурата за заемно споредување на сите кандидати за минимални пат вектори е релативно скапа, ако се земе предвид дека бројот на кандидатите за минимални пат вектори е многу поголем од бројот на јазли или од бројот на линкови во транспортниот систем. Поради тоа, сакаме да најдеме начин за определување дали даден кандидат за минимален пат вектор е минимален пат вектор без споредување на кандидат векторите. Според Теорема 4.1 за насочени транспортни системи, а според Теорема 4.2 и Теорема 4.3 за ненасочени транспортни системи, графот што е индуциран од минимален пат вектор е насочен ацикличен граф. Насочените ациклични графови може да се користат за моделирање на различни типови на проблеми. За нас ова е значајно бидејќи **релацијата на достапност** во насочен ацикличен граф дефинира подредување на јазлите.

Според кажаното погоре, за да провериме дали еден кандидат е минимален пат вектор, доволно е да провериме дали графот индуциран од соодветниот вектор е ацикличен. Во докторската дисертација во Глава 4 и Глава 5, проверката дали графот индуциран од соодветниот вектор е ацикличен се проверува со примена на DFS на остаточниот транспортен систем што се добива за соодветниот вектор. Во овој дел ние ќе дадеме нов начин на проверка дали соодветниот граф е ацикличен. За таа цел, ние овде ја формираме матрицата на соседство и со анализа на нејзините елементи проверуваме дали соодветниот граф ќе содржи циклуси. Нашата цел е колку што може побргу да ги елиминираме кандидатите за минимални пат вектори кои индуцираат графови што содржат циклуси. За релацијата на достапност е јасно дека е транзитивен затворац на релацијата соседство. Но процедурата за наоѓање транзитивен затворац, во општ случај

има голема сложеност, матрицата на соседство треба да се помножи сама со себе најмногу $|V|$ пати. Ние ќе покажеме дека транзитивниот затворац може да се добие со три множења на матрицата на соседство.

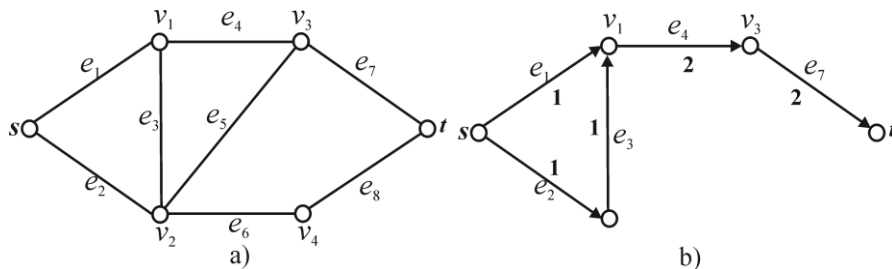
Нека е даден повеќе - состојбениот двотерминален транспортен систем $G(V, E, c)$. Да претпоставиме дека i -от линк може да работи во едно од нивоата од множеството $\{0, 1, \dots, M_i\}$, додека максималниот проток што може да се пренесе низ транспортниот систем да го означиме со M . ($\{0, 1, \dots, M\}$ е множество од сите можни протоци што може да се пренесат од изворот до сливникот).

Да претпоставиме дека транспортниот систем работи во состојба \vec{x} , каде \vec{x} е минимален пат вектор за ниво d . За да се пренесат d единици од изворот до сливникот, секој линк се користи само во една насока [38]. Да ја разгледаме само структурата на транспортниот систем, притоа да ги занемариме капацитетите на линковите, т.е. да го разгледуваме транспортниот систем како нетежински граф. Претпоставуваме дека линковите што се користат во \vec{x} се ориентирани во истата насока како и во \vec{x} , а останатите линкови (што не се користат во \vec{x}) се бришат од графот. Нека овој ацикличен граф го означиме со $G_{\vec{x}}(V, E_{\vec{x}})$. Соодветната матрица на достапност за $G_{\vec{x}}$ ќе ја означиме со $\Lambda_{\vec{x}}$. Бидејќи графот $G_{\vec{x}}(V, E_{\vec{x}})$ е ацикличен, можеме да дефинираме подредување на јазлите во однос на \vec{x} .

Дефиниција 5.1 Нека \vec{x} е минимален пат вектор за ниво d . За два јазли u и v , $u \rightarrow_{\vec{x}} v$ ако постои пат во G од u до v . Ваквиот тип на подредување ќе го наречеме подредување на јазлите во однос на \vec{x} .

Всушност, релацијата $\rightarrow_{\vec{x}}$ е **релација на достапност** за $G_{\vec{x}}$.

Пример 5.3 Да го разгледаме транспортниот систем даден на Фигура 5.3 а). Еден 2-МР е $(1, 1, 1, 2, 0, 0, 2, 0)$, како што е прикажано на Фигура 5.3 а). На Фигура 5.3 б) е прикажано подредувањето на јазлите во однос на векторот $\vec{x} = (1, 1, 1, 2, 0, 0, 2, 0)$.



Фигура 5.3. а) Ненасочена мрежа; б) Подредување на јазлите во однос на $\vec{x} = (1, 1, 1, 2, 0, 0, 2, 0)$.

Дефиниција 5.2 Нека \vec{x} и \vec{y} се пат вектори во G за ниво d и d' соодветно. Дефинираме граф $G_{\vec{x}+\vec{y}}(V, \tilde{E}_{\vec{x}+\vec{y}})$, каде $(u, v) \in \tilde{E}_{\vec{x}+\vec{y}} \Leftrightarrow (u, v) \in \tilde{E}_{\vec{x}}$ или $(u, v) \in \tilde{E}_{\vec{y}}$.

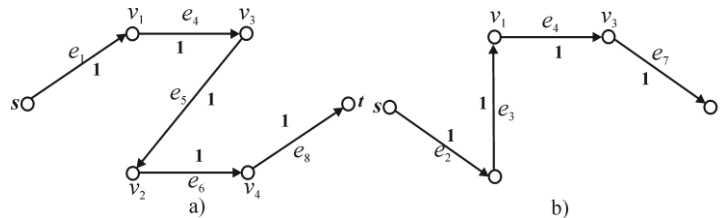
Како што покажавме во Глава 4, графот што е индуциран од минимален пат вектор е ацикличен. Значи, за да покажеме дали еден кандидат за минимален пат вектор е минимален пат вектор доволно е да провериме дали соодветниот граф е ацикличен. Тоа може да се направи со користење на матрица на достапност. Ако графот од кој е индуцирана матрицата на достапност е ацикличен тогаш сите нејзини дијагонални елементи се нули. Претходните алгоритми истото ова го правеа со DFS пребарување на остаточниот транспортен систем на графот што е индуциран од кандидат за минимален пат вектор. Причината поради која го предложуваме овој алгоритам е што во голем број случаи може да се провери за време $O(1)$ дали кандидатот за минимален пат вектор содржи циклус. Нека имаме два минимални пат вектори, \vec{x} и \vec{y} , нека $G_{\vec{x}}$ и $G_{\vec{y}}$ се графовите индуцирани од \vec{x} и \vec{y} , и $\Lambda_{\vec{x}}$ и $\Lambda_{\vec{y}}$ нека се соодветните матрици на достапност. Тогаш, доколку во едната матрица на достапност има единица на позицијата (i, j) (значи постои линк од v_i до v_j), а во другата матрица постои единица на позицијата (j, i) (значи постои линк од v_j до v_i) јасно е дека на дијагоналата на матрицата $\Lambda_{\vec{x}+\vec{y}}$ ќе има единица, па графот што е индуциран од векторот $\vec{x} + \vec{y}$ ќе содржи циклус, од каде следува дека $\vec{x} + \vec{y}$ не е минимален пат вектор. Доколку ова не може да се направи тогаш за да се забрза постапката подобро ќе биде да се примени некој од останите алгоритми што ги предлагаме во оваа глава. Поточно за соодветниот вектор ќе го формираме остаточниот транспортен системи, и со помош на DFS ќе провериме дали содржи циклус.

Следниве својства се доста значајни, бидејќи во општ случај процедурата за наоѓање транзитивен затворац има голема сложеност, т.е. за да се добие матрицата на достапност, матрицата на соседство треба да се помножи сама со себе најмногу $|V|$ пати. Ние ќе покажеме дека транзитивниот затворац а со тоа и матрицата на достапност може да се добие со три множења на матрицата на соседство. За таа цел ќе ја дефинираме релацијата $\tilde{\alpha}_{\vec{x}+\vec{y}}$ на следниот начин:

$$u \tilde{\alpha}_{\vec{x}+\vec{y}} v \Leftrightarrow u \xrightarrow{x} v \text{ или } u \xrightarrow{y} v, \quad (5.1)$$

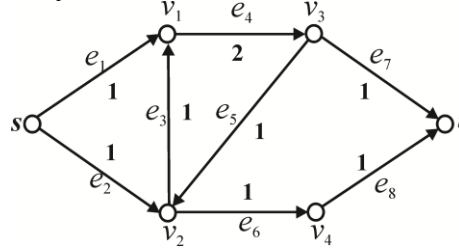
и релацијата $\alpha_{\vec{x}+\vec{y}}$ како транзитивен затворац на $\tilde{\alpha}_{\vec{x}+\vec{y}}$. Јасно е дека $\alpha_{\vec{x}+\vec{y}}$ е релација на достапност за $G_{\vec{x}+\vec{y}}$.

Пример 5.4 Векторите $(1, 0, 0, 1, 1, 1, 0, 1)$ и $(0, 1, 1, 1, 0, 0, 1, 0)$ се минимални пат вектори за транспортниот систем прикажан на Фигура 5.4 а).



Фигура 5.4. а) Минимален пат вектор за ниво 1, $(1, 0, 0, 1, 1, 1, 0, 1)$; б) Минимален пат вектор за ниво 1 $(0, 1, 1, 1, 0, 0, 1, 0)$.

Со собирање на овие вектори се добива векторот $(1, 1, 1, 2, 1, 1, 1, 1)$ кој не е минимален пат вектор за ниво 2. Овој вектор е поголем од векторот $(1, 1, 0, 1, 0, 1, 1, 1)$. На Фигура 5.5 е прикажано дека соодветниот граф што се добива со собирање на овие вектори содржи циклус. Како што докажавме и во Теорема 4.1, овој вектор не може да биде минимален пат вектор бидејќи содржи циклус.



Фигура 5.5 а) Пат вектор за ниво 2, $(1, 1, 1, 2, 1, 1, 1, 1)$ кој го содржи циклусот $\langle v_2, v_1, v_3, v_2 \rangle$.

Нека $\tilde{\Lambda}_{\vec{x}+\vec{y}} = \tilde{\Lambda}_{\vec{x}} \oplus \tilde{\Lambda}_{\vec{y}}$, каде \oplus е оператор за бинарно собирање ($1 \oplus 1 = 1, 1 \oplus 0 = 1, 0 \oplus 1 = 1, 0 \oplus 0 = 0$). Јасно е дека

$$\Lambda_{\vec{x}+\vec{y}} = \sum_{i=1}^{|V|} \tilde{\Lambda}_{\vec{x}+\vec{y}}^i. \quad (5.2)$$

каде собирањето е бинарно собирање (\oplus) додека множењето е бинарно множење ($1 \otimes 1 = 1, 1 \otimes 0 = 0, 0 \otimes 0 = 1, 0 \otimes 0 = 0$).

Да ги разгледаме векторите $\vec{x} = (1, 0, 0, 1, 1, 1, 0, 1)$ и $\vec{y} = (0, 1, 1, 1, 0, 0, 1, 0)$ кои се прикажани на Фигура 5.4. Со собирање на овие два минимални пат вектори се добива кандидатот за минимален пат вектор $(1, 1, 1, 2, 1, 1, 1, 1)$, кој не е минимален. Графот што е индуциран од овој вектор го содржи циклусот $\langle v_2, v_1, v_3, v_2 \rangle$, како што е прикажано на Фигура 5.5. Но ако ги разгледаме матриците на достапност во време $O(1)$ без да ги собираме векторите, може да заклучиме дека кандидатот не е минимален пат вектор. Матриците на достапност се следниве:

$$\Lambda_{\vec{x}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \Lambda_{\vec{y}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

Можеме да забележиме дека матрицата $\Lambda_{\vec{x}}$ на позиција $(2, 3)$ има единица, додека матрицата $\Lambda_{\vec{y}}$ има единица на позиција $(3, 2)$. Врз основа на ова можеме да заклучиме дека векторот што се добива со собирање на векторите \vec{x} и \vec{y} нема да биде минимален

пат вектор. Значи, до овој заклучок можеме да дојдеме без да ја најдеме матрицата на достапност $\Lambda_{\vec{x}+\vec{y}}$.

Нека $G(V, E)$ е соодветниот граф за дадениот повеќе - состојбен транспортен систем. За секој прост пат $\langle s = u_0, u_1, u_2, \dots, u_{k-2}, u_{k-1}, u_k = t \rangle$, $(u_i, u_{i+1}) \in E$ ќе конструираме бинарен минимален пат вектор со ставање на 1 на соодветната координата за линкот (u_i, u_{i+1}) , ако u_i и u_{i+1} се елементи на простиот пат и на 0 на соодветната координата за линкот (u_i, u_{i+1}) , ако u_i или u_{i+1} не се елементи од патот. Всушност бинарен минимален пат за повеќе - состојбен транспортен систем е минимален пат вектор за бинарниот транспортен систем со ист нетежински граф. На пример, соодветниот бинарен минимален пат вектор за простиот пат $\langle s, v_1, v_3, t \rangle$ за транспортниот систем од Фигура 5.3 а) е $(1, 0, 0, 1, 0, 0, 1, 0)$. Јасно е дека јазлите што лежат на простиот пат што одговара на \vec{x} се линеарно подредени во однос на \rightarrow_x . Ова ќе го искористиме за докажување на следново својство:

Својство 5.1 Нека \vec{x} е бинарен минимален пат вектор и $u, v, u_1, v_1 \in V$, тогаш од $u \rightarrow_x u_1$ (или $u_1 \rightarrow_x u$) и $v \rightarrow_x v_1$ (или $v_1 \rightarrow_x v$) следува дека или $u \rightarrow_x v$ или $v \rightarrow_x u$.

Доказ: Од $u \rightarrow_x u_1$ следува дека u лежи на простиот пат што одговара на \vec{x} . Исто така, и u_1 лежи на простиот пат што одговара на \vec{x} па u и u_1 се линеарно подредени во однос на \vec{x} . Истото важи и за јазлите v и v_1 . Значи сите јазли што лежат на простиот пат што одговара на \vec{x} се линеарно подредени во однос на \rightarrow_x , имаме дека или $u \rightarrow_x v$ или $v \rightarrow_x u$. □

Резултатите од следново својство и последицата што произлегува од него се корисни за намалување на бројот на чекори за определување на релацијата на достапност на графот $G_{\vec{x}+\vec{y}}$.

Својство 5.2 Нека \vec{x} е минимален пат вектор за ниво d , \vec{y} е бинарен минимален пат вектор и u, v, w и r се јазли така што $u \rightarrow_y v \rightarrow_x w \rightarrow_y r$. Тогаш или $u \rightarrow_y r$ или најмалку еден елемент на дијагоналата на матрицата $\tilde{L}_{\vec{x}+\vec{y}}^2$ е еднаков на 1.

Доказ: Нека $u \rightarrow_y v \rightarrow_x w \rightarrow_y r$ и \vec{y} е бинарен минимален пат вектор. Од Својство 5.1 имаме дека или $u \rightarrow_y w$ или $w \rightarrow_y u$.

Ако $u \rightarrow_y w$, тогаш $u \rightarrow_y w \rightarrow_y r$ и бидејќи \rightarrow_y е транзитивна релација следува дека $u \rightarrow_y r$.

Ако $w \xrightarrow{y} u$, тогаш $w \xrightarrow{y} u \xrightarrow{y} v$ и бидејќи \xrightarrow{y} е транзитивна релација следува дека $w \xrightarrow{y} v$. Имаме, $w \xrightarrow{y} v$ и $v \xrightarrow{x} w$ од каде $w\tilde{\alpha}_{\bar{x}+\bar{y}}v$ и $v\tilde{\alpha}_{\bar{x}+\bar{y}}w \Rightarrow w\tilde{\alpha}_{\bar{x}+\bar{y}}^2w$, т.е. на дијагоналата на матрицата $\tilde{\Lambda}_{\bar{x}+\bar{y}}^2$ најмалку еден елемент е еднаков на 1. \square

Последица 5.1 Нека \bar{x} е минимален пат вектор за ниво d , \bar{y} е бинарен минимален пат вектор и u, v, w, r и m се јазли, така што $u \xrightarrow{x} v \xrightarrow{y} w \xrightarrow{x} r \xrightarrow{y} m$. Тогаш или $u\tilde{\alpha}_{\bar{x}+\bar{y}}^2m$ или најмалку еден елемент на дијагоналата на матрицата $\tilde{\Lambda}_{\bar{x}+\bar{y}}^2$ е еднаков на 1.

Доказ: Ако на дадениот услов $v \xrightarrow{y} w \xrightarrow{x} r \xrightarrow{y} m$ се примени Својство 5.2 се добива дека или најмалку еден елемент на дијагоналата на матрицата $\tilde{\Lambda}_{\bar{x}+\bar{y}}^2$ е еднаков на 1 или $v \xrightarrow{y} m$. Бидејќи $v \xrightarrow{y} m$ и од дадениот услов имаме дека $u \xrightarrow{x} v$ следува дека $u \xrightarrow{x} v \xrightarrow{y} m \Rightarrow u\tilde{\alpha}_{\bar{x}+\bar{y}}^2m$. \square

Следниот резултат покажува дека сите циклуси можат да бидат најдени со споредба на елементите од матриците $\Lambda_{\bar{x}}$ и $\Lambda_{\bar{y}}$.

Теорема 5.1 Ако $G_{\bar{x}+\bar{y}}$ има циклус, тогаш постои јазол u така што $u\tilde{\alpha}_{\bar{x}+\bar{y}}^2u$ или $u\tilde{\alpha}_{\bar{x}+\bar{y}}^3u$.

Доказ: Да го претпоставиме спротивното, т.е. нека $k > 3$ е најмалиот цел број за кој $w\tilde{\alpha}_{\bar{x}+\bar{y}}^kw$. Тогаш постојат јазли $w_i, i = 0, 1, \dots, k$ така што:

$$w\tilde{\alpha}_{\bar{x}+\bar{y}}w_0\tilde{\alpha}_{\bar{x}+\bar{y}}w_1\tilde{\alpha}_{\bar{x}+\bar{y}}\dots\tilde{\alpha}_{\bar{x}+\bar{y}}w_{k-1}\tilde{\alpha}_{\bar{x}+\bar{y}}w_k = w. \quad (5.3)$$

Поради тоа што поднизата од (5.3) има најмалку 4 елементи, имаме дека постои подниза $w_j \xrightarrow{y} w_{j+1} \xrightarrow{x} w_{j+2} \xrightarrow{y} w_{j+3}$. Со примена на Својство 5.2 или $w_j\tilde{\alpha}_{\bar{x}+\bar{y}}^2w_j$ или $w_j \xrightarrow{y} w_{j+3}$. И двата случаја се во кондрадикција со претпоставката дека $k > 3$ е најмалиот цел број така што $w\tilde{\alpha}_{\bar{x}+\bar{y}}^kw$. \square

Сега сме спремни да докажеме дека матрицата на достапност за графот што е добиен со додавање на нов прост пат \bar{y} на графот $G_{\bar{x}}$ може да се определи во 3 чекори.

Теорема 5.2 Нека \vec{x} е минимален пат вектор за ниво d , \vec{y} е бинарен минимален пат вектор и $G_{\vec{x}+\vec{y}}$ не содржи циклуси. Тогаш

$$\tilde{\alpha}_{\vec{x}+\vec{y}}^4 \subseteq \tilde{\alpha}_{\vec{x}+\vec{y}} \cup \tilde{\alpha}_{\vec{x}+\vec{y}}^2 \cup \tilde{\alpha}_{\vec{x}+\vec{y}}^3. \quad (5.4)$$

Доказ: Нека $u\tilde{\alpha}_{\vec{x}+\vec{y}}^2v$, тогаш имаме

$$\begin{aligned} u\tilde{\alpha}_{\vec{x}+\vec{y}}^2v &\Rightarrow (\exists w \in V)u\tilde{\alpha}_{\vec{x}+\vec{y}}w\tilde{\alpha}_{\vec{x}+\vec{y}}v \\ &\Rightarrow (\exists w \in V)(u \xrightarrow{x} w \vee u \xrightarrow{y} w) \wedge (w \xrightarrow{x} v \vee w \xrightarrow{y} v) \\ &\Rightarrow (\exists w \in V)(u \xrightarrow{x} w \xrightarrow{x} v) \vee (u \xrightarrow{y} w \xrightarrow{y} v) \\ &\quad \vee (u \xrightarrow{x} w \xrightarrow{y} v) \vee (u \xrightarrow{y} w \xrightarrow{x} v). \end{aligned}$$

Релациите \xrightarrow{x} и \xrightarrow{y} се транзитивни, па според тоа

$$(\exists w \in V)(u \xrightarrow{x} v) \vee (u \xrightarrow{y} w \xrightarrow{y} v) \vee (u \xrightarrow{x} w \xrightarrow{y} v) \vee (u \xrightarrow{y} w \xrightarrow{x} v). \quad (5.5)$$

Нека $u\tilde{\alpha}_{\vec{x}+\vec{y}}^3v$, од (5.5) имаме

$$\begin{aligned} u\tilde{\alpha}_{\vec{x}+\vec{y}}^3v &\Rightarrow (\exists w_1 \in V)(u\tilde{\alpha}_{\vec{x}+\vec{y}}^2w_1) \wedge (w_1\tilde{\alpha}_{\vec{x}+\vec{y}}v) \\ &\Rightarrow (\exists w_1 \in V)((\exists w \in V)((u \xrightarrow{x} w_1) \vee (u \xrightarrow{y} w \xrightarrow{x} w_1) \\ &\quad \vee (u \xrightarrow{x} w \xrightarrow{y} w_1) \vee (u \xrightarrow{y} w_1)) \wedge ((w_1 \xrightarrow{x} v) \vee (w_1 \xrightarrow{y} v))) \\ &\Rightarrow (\exists w, w_1 \in V)((u \xrightarrow{x} w_1 \xrightarrow{x} v) \vee (u \xrightarrow{x} w_1 \xrightarrow{y} v) \\ &\quad \vee (u \xrightarrow{y} w \xrightarrow{x} w_1 \xrightarrow{x} v) \vee (u \xrightarrow{y} w \xrightarrow{x} w_1 \xrightarrow{y} v) \\ &\quad \vee (u \xrightarrow{x} w \xrightarrow{y} w_1 \xrightarrow{x} v) \vee (u \xrightarrow{x} w \xrightarrow{y} w_1 \xrightarrow{y} v) \\ &\quad \vee (u \xrightarrow{y} w_1 \xrightarrow{x} v) \vee (u \xrightarrow{y} w_1 \xrightarrow{y} v)). \end{aligned}$$

Релациите \xrightarrow{x} и \xrightarrow{y} се транзитивни, па според тоа

$$\begin{aligned} (\exists w, w_1 \in V)((u \xrightarrow{x} v) \vee (u \xrightarrow{x} w_1 \xrightarrow{x} v) \vee (u \xrightarrow{y} w \xrightarrow{x} v) \\ \vee (u \xrightarrow{y} w \xrightarrow{x} w_1 \xrightarrow{x} v) \vee (u \xrightarrow{x} w \xrightarrow{y} w_1 \xrightarrow{x} v) \\ \vee (u \xrightarrow{x} w \xrightarrow{y} v) \vee (u \xrightarrow{y} w_1 \xrightarrow{x} v) \vee (u \xrightarrow{y} v)). \end{aligned}$$

Ако имаме предвид дека нема циклуси во $G_{\vec{x}+\vec{y}}$, со користење на Својство 5.2 може да ја направиме следната редукција

$$\begin{aligned} (\exists w, w_1 \in V) & ((u \xrightarrow{x} v) \vee (u \xrightarrow{x} w_1 \xrightarrow{y} v) \vee (u \xrightarrow{x} w \xrightarrow{y} v) \vee (u \xrightarrow{y} v) \\ & \vee (u \xrightarrow{x} w \xrightarrow{y} w_1 \xrightarrow{x} v) \vee (u \xrightarrow{x} w \xrightarrow{y} v) \vee (u \xrightarrow{y} w_1 \xrightarrow{x} v)). \end{aligned}$$

Според тоа,

$$(\exists w, w_1 \in V) ((u \tilde{\alpha}_{\vec{x}+\vec{y}}^2 v) \vee (u \xrightarrow{x} w \xrightarrow{y} w_1 \xrightarrow{x} v)). \quad (5.6)$$

На крај нека $(u \tilde{\alpha}_{\vec{x}+\vec{y}}^4 v)$. Од (5.5) и (5.6) имаме

$$\begin{aligned} (u \tilde{\alpha}_{\vec{x}+\vec{y}}^4 v) & \Rightarrow (\exists w_2 \in V) (u \tilde{\alpha}_{\vec{x}+\vec{y}}^3 w_2) \wedge (w_2 \tilde{\alpha}_{\vec{x}+\vec{y}} v) \\ & \Rightarrow (\exists w, w_1, w_2 \in V) ((u \tilde{\alpha}_{\vec{x}+\vec{y}}^3 w_2) \vee (u \xrightarrow{x} w \xrightarrow{y} w_1 \xrightarrow{x} w_2)) \\ & \quad \wedge ((w_2 \xrightarrow{x} v) \vee (w_2 \xrightarrow{y} v)) \\ & \Rightarrow (\exists w, w_1, w_2 \in V) ((u \tilde{\alpha}_{\vec{x}+\vec{y}}^3 v) \vee (u \xrightarrow{x} w \xrightarrow{y} w_1 \xrightarrow{x} v) \\ & \quad \vee (u \xrightarrow{x} w \xrightarrow{y} w_1 \xrightarrow{x} w_2 \xrightarrow{y} v)). \end{aligned}$$

Со користење на Својството 5.2 и фактот дека нема циклуси во $G_{\vec{x}+\vec{y}}$, добиваме

$$u \tilde{\alpha}_{\vec{x}+\vec{y}}^4 v \Rightarrow u \tilde{\alpha}_{\vec{x}+\vec{y}}^3 v \vee u \tilde{\alpha}_{\vec{x}+\vec{y}}^2 v, \quad (5.7)$$

од каде следува (5.4)

$$\tilde{\alpha}_{\vec{x}+\vec{y}}^4 \subseteq \tilde{\alpha}_{\vec{x}+\vec{y}} \cup \tilde{\alpha}_{\vec{x}+\vec{y}}^2 \cup \tilde{\alpha}_{\vec{x}+\vec{y}}^3. \quad \square$$

Следниот резултат следува директно од претходната теорема.

Последица 5.2 Нека \vec{x} е минимален пат вектор за ниво d , \vec{y} е бинарен минимален пат вектор и $G_{\vec{x}+\vec{y}}$ не содржи циклуси, тогаш $\Lambda_{\vec{x}+\vec{y}} = \tilde{\Lambda}_{\vec{x}+\vec{y}} \oplus \tilde{\Lambda}_{\vec{x}+\vec{y}}^2 \oplus \tilde{\Lambda}_{\vec{x}+\vec{y}}^3$.

Со користење на претходните резултати, ќе предложиме алгоритам за наоѓање на минимални пат вектори за ниво d (d -MP) на повеќе - состојбен транспортен систем.

Алгоритам 5.3 – Алгоритам со контрола на циклуси ($G(V, E, c)$)

Чекор 1. Се наоѓаат сите бинарни минимални пат вектори.

Чекор 2. За секој бинарен минимален пат вектор \vec{x} се наоѓа матрицата $\Lambda_{\vec{x}}$.

Чекор 3. Се иницијализира $d+1\text{-}MP' = \emptyset$.

Чекор 4. За секој минимален пат вектор за ниво d , \vec{x} и секој бинарен минимален пат вектор \vec{y} најди ја матрицата $\Lambda_{\vec{x}+\vec{y}}$.

Чекор 5. $\Lambda_{\vec{x}+\vec{y}} = \tilde{\Lambda}_{\vec{x}+\vec{y}} \oplus \tilde{\Lambda}_{\vec{x}+\vec{y}}^2 \oplus \tilde{\Lambda}_{\vec{x}+\vec{y}}^3$.

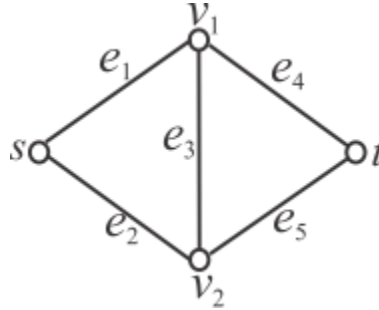
Чекор 6. Ако сите дијагонални елементи на $\Lambda_{\vec{x}+\vec{y}}$ се нула, тогаш $d+1\text{-}MP' = d+1\text{-}MP' \cup \{\vec{x} + \vec{y}\}$.

Чекор 7. Множеството $d+1\text{-}MP$ се добива од множеството $d+1\text{-}MP'$ со бришење на елементите што се појавуваат повеќе од еднаш.

Чекор 8. Се повторуваат чекорите 3, 4, 5 и 6 за секој $d \leq M$.

За да видиме како работи алгоритмот, ќе го разгледаме следниот пример.

Пример 5.6 Да ја разгледаме мрежата дадена на Фигура 5.6



Фигура 5.6 Ненасочен транспортен систем

Чекор 1. Бинарни минимални пат вектори се $\vec{x}_1 = (1, 0, 1, 0, 1)$, $\vec{x}_2 = (1, 0, 0, 1, 0)$, $\vec{x}_3 = (0, 1, 1, 1, 0)$, $\vec{x}_4 = (0, 1, 0, 0, 1)$.

Чекор 2. Соодветните матрици на достапност се:

$$\Lambda_{\vec{x}_1} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \Lambda_{\vec{x}_2} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \Lambda_{\vec{x}_3} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \Lambda_{\vec{x}_4} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Чекор 3. Се иницијализира $2\text{-}MP' = \emptyset$.

Чекор 4. За сите минимални пат вектори во 1- MP се наоѓаат матриците $\tilde{\Lambda}_{\vec{x}_k + \vec{x}_l}$, $k, l = 1, 2, 3, 4$. На пример:

$$\tilde{\Lambda}_{\vec{x}_1 + \vec{x}_1} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{\Lambda}_{\vec{x}_1 + \vec{x}_2} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{\Lambda}_{\vec{x}_1 + \vec{x}_3} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

На ист начин се добиваат сите матрици $\tilde{\Lambda}_{\vec{x}_k + \vec{x}_l}$, $k, l = 1, 2, 3, 4$.

Чекор 5. $\Lambda_{\vec{x} + \vec{y}} = \tilde{\Lambda}_{\vec{x} + \vec{y}} \oplus \tilde{\Lambda}_{\vec{x} + \vec{y}}^2 \oplus \tilde{\Lambda}_{\vec{x} + \vec{y}}^3$.

За векторот $\vec{x}_1 + \vec{x}_1$, се добива:

$$\tilde{\Lambda}_{\vec{x}_1 + \vec{x}_1}^2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \tilde{\Lambda}_{\vec{x}_1 + \vec{x}_1}^3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\Lambda_{\vec{x}_1 + \vec{x}_1} = \tilde{\Lambda}_{\vec{x}_1 + \vec{x}_1} \oplus \tilde{\Lambda}_{\vec{x}_1 + \vec{x}_1}^2 \oplus \tilde{\Lambda}_{\vec{x}_1 + \vec{x}_1}^3 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

За векторот $\vec{x}_1 + \vec{x}_3$, се добива

$$\tilde{\Lambda}_{\vec{x}_1 + \vec{x}_3}^2 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \tilde{\Lambda}_{\vec{x}_1 + \vec{x}_3}^3 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\Lambda_{\vec{x}_1 + \vec{x}_3} = \tilde{\Lambda}_{\vec{x}_1 + \vec{x}_3} \oplus \tilde{\Lambda}_{\vec{x}_1 + \vec{x}_3}^2 \oplus \tilde{\Lambda}_{\vec{x}_1 + \vec{x}_3}^3 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Чекор 6. Во матриците на достапност проверуваме дали постојат дијагонални елементи еднакви на 1. Овде треба да забележиме дека матрицата $\Lambda_{\vec{x}_1 + \vec{x}_3}$ има дијагонални елементи еднакви на 1, па векторот $\vec{x}_1 + \vec{x}_3 = (1, 1, 2, 1, 1)$ не е минимален пат вектор за ниво 2.

Останатите минимални пат вектори за ниво 2 заедно со нивните матрици на достапност се дадени во Табела 5.1.

Табела 5. 1 Минимални пат вектори за ниво 2.

1-МР	1-МР	$\tilde{\Lambda}_{\vec{x}_k + \vec{x}_l}$	2-МР
(1, 0, 1, 0, 1)	(1, 0, 1, 0, 1)	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	(2, 0, 2, 0, 2)
(1, 0, 1, 0, 1)	(1, 0, 0, 1, 0)	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	(2, 0, 1, 1, 1)
(1, 0, 1, 0, 1)	(0, 1, 1, 1, 0)	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	(1, 1, 2, 1, 1) не припаѓа во 2-МР

(1, 0, 1, 0, 1)	(0, 1, 0, 0, 1)	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	(1, 1, 1, 0, 2)
(1, 0, 0, 1, 0)	(1, 0, 0, 1, 0)	$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	(2, 0, 0, 2, 0)
(1, 0, 0, 1, 0)	(0, 1, 1, 1, 0)	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	(1, 1, 1, 2, 0)
(1, 0, 0, 1, 0)	(0, 1, 0, 0, 1)	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	(1, 1, 0, 1, 1)
(0, 1, 1, 1, 0)	(0, 1, 1, 1, 0)	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	(0, 2, 2, 2, 0)
(0, 1, 1, 1, 0)	(0, 1, 0, 0, 1)	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	(0, 2, 1, 1, 1)
(0, 1, 0, 0, 1)	(0, 1, 0, 0, 1)	$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	(0, 2, 0, 0, 2)

Во предложениот алгоритам, секој кандидат за минимален пат вектор се проверува дали навистина е минимален пат вектор со $O(|V|^3)$ бинарни операции. Предноста на овој алгоритам во однос на алгоритмите предложени во [19], [21], [36], [45] и [60] е што одзема помалку време отколку операцијата споредување на вектори (оваа операција има временска комплексност $O(\lambda^2|E|)$, каде λ е бројот на кандидати за минимални пат вектори) што се користи за елиминација на кандидатите што не се минимални пат вектори. Меѓутоа, треба да нагласиме дека во останатите алгоритми предложени во оваа глава, проверката дали постои циклус се врши со DFS, кој има временска сложеност $O(E)$. Предноста на алгоритмот што го даваме во овој дел, најмногу се состои во тоа што постојат случаи кога не мора да ги собираме векторите, поточно можеме да заклучиме

дали постои циклус во соодветниот граф, врз база на тоа дали има единица на позицијата (i, j) во едната матрица и на позицијата (j, i) во другата матрица на достапност.

5.2 Алгоритми за наоѓање на минимални пат вектори за насочени транспортни системи со користење на циклуси во остаточниот транспортен систем

Во подглава 4.1 ја дадовме врската помеѓу минималните пат вектори за ниво d и протоците со вредност d . Следно, сакаме да ја најдеме врската помеѓу два минимални пат вектори за ниво d . Со ова ќе предложиме нов пристап со чија помош ќе можеме да го најдеме целото множество од минимални пат вектори за ниво d ако знаеме само еден минимален пат вектор за ниво d . Да нагласиме дека со алгоритмот на Ford - Fulkerson може да се најде максималниот проток во даден двотерминален транспортен систем. Заедно со протокот го наоѓаме и соодветниот остаточен транспортен систем. Временска комплексност на алгоритмот на Ford-Fulkerson е $O(|V||E|^2)$. Со мала модификација на алгоритмот на Ford-Fulkerson може да најдеме и еден проток со вредност d . Откако ќе најдеме еден проток со вредност d со примена на Теорема 4.1, можеме да видиме дали векторот индуциран од овој проток е минимален пат вектор (со едноставна проверка дали графот индуциран од проток е ацикличен, ние можеме да најдеме еден минимален пат вектор за ниво d .)

Следната теорема ја дава врската помеѓу два протоци. Ова ќе ни помогне да предложиме друг алгоритам за пресметување на сите d -MPs.

Теорема 5.3 Нека f е проток со $|f| = d$ во двотерминален проточен транспортен систем $G(V, E, c)$ со извор s и сливник t , и \mathcal{C} е зголемувачки циклус во остаточниот граф $G_f(V, E_f)$. За $d' \leq \min\{c_f(e_i) \mid e_i \in \mathcal{C}\}$, функцијата f_1 дефинирана со

$$f_1(u, v) = \begin{cases} f(u, v) + d', & (u, v) \in \mathcal{C} \cap E \\ f(u, v) - d', & (v, u) \in \mathcal{C} \text{ и } (u, v) \in E, \\ f(u, v), & (u, v) \notin \mathcal{C} \text{ и } (v, u) \notin \mathcal{C} \end{cases} \quad (5.8)$$

е проток во $G(V, E, c)$ со $|f_1| = |f|$.

Доказ: За да се покаже дека f_1 е проток ние треба да покажеме дека правилата за ограничување на капацитетот и за ограничување за одржување на протокот се задоволени.

Ограничување на капацитетот:

- Ако $(u, v) \notin \mathcal{C}$ и $(v, u) \notin \mathcal{C}$, имаме $f_1(u, v) = f(u, v)$ од каде следува $0 \leq f_1(u, v) \leq c(u, v)$.
- Ако $(u, v) \in \mathcal{C} \cap E$ имаме $f_1(u, v) = f(u, v) + d' \geq f(u, v) \geq 0$ и $f_1(u, v) = f(u, v) + d' \leq f(u, v) + c_f(u, v) = f(u, v) + c(u, v) - f(u, v) = c(u, v)$.
- Ако $(v, u) \in \mathcal{C}$ и $(u, v) \in E$, $f_1(u, v) = f(u, v) - d' \leq f(u, v) \leq c(u, v)$ и

$$f_1(u, v) = f(u, v) - d' \geq f(u, v) - c_f(u, v) = f(u, v) - f(u, v) \geq 0.$$

Одржување на протокот:

За да се покаже дека условот за одржување на протокот е задоволен, доволно е да се покаже дека

$$\sum_{u \in V} f_1(u, v) - \sum_{w \in V} f_1(v, w) = \sum_{u \in V} f(u, v) - \sum_{w \in V} f(v, w). \quad (5.9)$$

Јасно, последната равенка важи за $v \notin \mathcal{C}$, бидејќи во овој случај $f(u, v) = f_1(u, v)$ и $f(v, w) = f_1(v, w)$, за сите u и w .

Ако $u \in \mathcal{C}$, бидејќи \mathcal{C} е прост циклус, тогаш u може да се појави точно еднаш во \mathcal{C} . Нека u_1 и w_1 се јазли така што $(u_1, v) \in \mathcal{C}$ и $(v, w_1) \in \mathcal{C}$. Постојат четири можности:

- За $(u_1, v) \in E$ и $(v, w_1) \in E$

$$\begin{aligned} f_1(V, v) - f_1(v, V) &= f_1(V \setminus \{u_1\}, v) + f_1(u_1, v) - f_1(v, V \setminus \{w_1\}) - f_1(v, w_1) \\ &= f(V \setminus \{u_1\}, v) + f(u_1, v) + d' - f(v, V \setminus \{w_1\}) - (f(v, w_1) + d') \\ &= f(V, v) - f(v, V). \end{aligned}$$
- За $(u_1, v) \in E$ и $(w_1, v) \in E$

$$\begin{aligned} f_1(V, v) - f_1(v, V) &= f_1(V \setminus \{u_1, w_1\}, v) + f_1(u_1, v) + f_1(w_1, v) - f_1(v, V) \\ &= f(V \setminus \{u_1, w_1\}, v) + f(u_1, v) + d' + f(w_1, v) - d' - f(v, V) \\ &= f(V, v) - f(v, V). \end{aligned}$$
- За $(v, u_1) \in E$ и $(v, w_1) \in E$

$$\begin{aligned} f_1(V, v) - f_1(v, V) &= f_1(V, v) - f_1(v, V \setminus \{u_1, w_1\}) - f_1(v, u_1) - f_1(v, w_1) \\ &= f(V, v) - f_1(v, V \setminus \{u_1, w_1\}) - (f(v, u_1) - d') - (f(v, w_1) + d') \\ &= f(V, v) - f(v, V). \end{aligned}$$
- За $(v, u_1) \in E$ и $(w_1, v) \in E$

$$\begin{aligned} f_1(V, v) - f_1(v, V) &= f_1(V \setminus \{w_1\}, v) + f_1(w_1, v) - f_1(v, V \setminus \{u_1\}) - f_1(v, u_1) \\ &= f(V \setminus \{w_1\}, v) + f(w_1, v) - d' - f(v, V \setminus \{u_1\}) - (f(v, u_1) - d') \\ &= f(V, v) - f(v, V). \end{aligned}$$

Доказот се комплетира со

$$|f_1| = \sum_{u \in V} f_1(u, s) - \sum_{w \in V} f_1(s, w) = \sum_{u \in V} f(u, s) - \sum_{w \in V} f(s, w) = |f|. \quad (5.10)$$

□

Директна последица од Теорема 5.3 е:

Последица 5.3 Нека \vec{x} е вектор од состојби за двотерминалниот проточен транспортен систем $G(V, E, c)$ со извор s и сливник t , псевдопротокот $l_{\vec{x}}$ индуциран од \vec{x} е проток со $|l_{\vec{x}}| = d$, и \vec{y} е зголемувачки вектор за ниво d' за циклусот \mathcal{C} во остаточниот транспортен систем $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$. Тогаш $\vec{x} + \vec{y}$ е вектор од состојби за $G(V, E, c)$, за кој псевдопротокот $l_{\vec{x}+\vec{y}}$ индуциран од $\vec{x} + \vec{y}$ е проток со $|l_{\vec{x}+\vec{y}}| = d$.

Лема 5.1 Нека f е проток со $|f| = 0$ во двотерминален транспортен $G(V, E, c)$, така што постои линк (u, v) за кој $f(u, v) > 0$. Тогаш графот $G(V, E^f)$ индуциран од f содржи циклус.

Доказ: Доказот следи директно од два факти. Првиот факт е дека влезниот степен на секој јазол во графот $G(V, E^f)$ е стриктно поголем од 0 ако и само ако нивниот излезен степен е исто така стриктно поголем од 0. Вториот факт е дека постои пат што поминува низ (u, v) . \square

Лема 5.2 Нека \vec{x} е вектор од состојби за двотерминалниот проточен транспортен систем $G(V, E, c)$ со извор s и сливник t и $l_{\vec{x}}$ е проток со $|l_{\vec{x}}| = 0$. Тогаш постојат зголемувачки вектори \vec{y}_k , за $k = 1, 2, \dots, r$ за ниво d'_k за циклусите \mathcal{C}_k во $G(V, E, c)$, така што $\vec{x} = \sum_{k=1}^r \vec{y}_k$.

Доказ: Бидејќи $l_{\vec{x}}$ е проток со $|l_{\vec{x}}| = 0$, од Лема 6.1 следува дека $G(V, E^{l_{\vec{x}}})$ го содржи циклусот \mathcal{C}_1 . Ако земеме $d'_1 = \min\{x_i \mid e_i \in \mathcal{C}_1\}$, можеме да формираме зголемувачки вектор \vec{y}_1 за ниво d'_1 . Векторот $\vec{z}_1 = \vec{x} - \vec{y}_1$ е вектор од состојби за двотерминалниот проточен транспортен систем $G(V, E, c)$ со $|l_{\vec{z}_1}| = 0$. Исто така, графот $G(V, E^{l_{\vec{z}_1}})$ има најмалку еден позитивен линк помал од соодветниот линк во графот $G(V, E^{l_{\vec{x}}})$. Ако во $G(V, E^{l_{\vec{z}_1}})$ не постои линк (u, v) за кој $l_{\vec{z}_1}(u, v) > 0$, следува дека доказот е завршен.

Во спротивно, продолжуваме со оваа процедура и ги формираме зголемувачкиот вектор \vec{y}_k за ниво d'_k за $G(V, E^{l_{\vec{z}_{k-1}}})$ и векторот $\vec{z}_k = \vec{x} - \vec{y}_k$. Со оваа процедура продолжуваме сè додека не добиеме дека $\vec{z}_k = \vec{0}$. Процедурата ќе заврши во најмалку $|E|$ чекори. Секој циклус \mathcal{C}_k е циклус во графот $G(V, E, c)$ бидејќи графот $G(V, E^{l_{\vec{z}_{k-1}}})$ е подграф на графот G . \square

Следната теорема покажува дека секој вектор индуциран од проток може да се добие со додавање на циклуси од $G(V, E^{l_{\vec{x}}})$ на векторот од состојби \vec{x} .

Теорема 5.4. Нека \vec{x} и \vec{y} се два вектори од состојби за проточниот транспортен систем $G(V, E, c)$ со извор s и сливник t , $l_{\vec{x}}$ и $l_{\vec{y}}$ се протоци со $|l_{\vec{x}}| = |l_{\vec{y}}| = d$. Тогаш постојат

зголемувачки вектори \vec{y}_k , за $k = 1, 2, \dots, r$ за ниво d'_k за циклусите \mathcal{C}_k во остаточниот транспортен систем $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$, така што $\vec{y} = \vec{x} + \sum_{k=1}^r \vec{y}_k$.

Доказ: Функцијата $-l_{\vec{x}}$ е проток во $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$ од t до s со $|l_{\vec{y}}| = d$ и нејзиниот остаточен граф е $G(V, E, c)$. Со користење на ова и фактот дека $l_{\vec{y}}$ е проток од s до t во $G(V, E, c)$ со $|l_{\vec{y}}| = d$, имаме дека функцијата $l_{\vec{y}-\vec{x}}$ дефинирана со $l_{\vec{y}-\vec{x}}(u, v) = l_{\vec{y}}(u, v) - l_{\vec{x}}(u, v)$ е проток во остаточниот граф $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$ со $|l_{\vec{y}-\vec{x}}(u, v)| = 0$. Векторот од состојби за $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$ индуциран од $l_{\vec{y}-\vec{x}}(u, v)$ е $\vec{y} - \vec{x}$. Бидејќи $|l_{\vec{y}-\vec{x}}(u, v)| = 0$, постојат зголемувачки вектори \vec{y}_k , за $k = 1, 2, \dots, r$ за ниво d'_k за циклусите \mathcal{C}_k во $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$ така што $\vec{y} - \vec{x} = \sum_{k=1}^r \vec{y}_k$. (од Лема 5.2). \square

Теорема 5.1, Теорема 5.3 и Теорема 5.4 се сублимирани во следнава теорема.

Теорема 5.5 Нека \vec{x} е d -MP во двотерминален проточен транспортен систем $G(V, E, c)$ со извор s и сливник t . Тогаш \vec{y} е d -MP ако и само ако $G(V, E^{l_{\vec{y}}})$ е ацикличен граф и постојат зголемувачки вектори \vec{y}_k , за $k = 1, 2, \dots, r$ за ниво d'_k за циклусите \mathcal{C}_k во остаточниот транспортен систем $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$, така што $\vec{y} = \vec{x} + \sum_{k=1}^r \vec{y}_k$.

Може да заклучиме дека Теорема 5.5 ни ја дава директно врската помеѓу два минимални пат вектори за исто ниво. Поточно, доколку најдеме еден минимален пат вектор останатите пат вектори ќе ги најдеме како збир од тој вектор и зголемувачките вектори за циклусите во остаточниот транспортен систем што се добива за најдениот минимален пат вектор. Претпоставуваме дека i -тата компонента може да работи во едно од нивоата од множеството $\{0, 1, \dots, M_i\}$, додека транспортниот систем може да работи во едно од нивоата од множеството $\{0, 1, \dots, M\}$. Значи M е максималниот проток што може да се пренесе во транспортниот систем. Со користење на Теорема 5.3, добиваме нов пристап кој ни помага при дизајнирањето на алгоритам за пресметување на сите d -MPs за дадено ниво $d \leq M$. Псевдокодот за алгоритам е даден подолу:

Алгоритам 5.4 – Алгоритам за наоѓање на минимални пат вектори со користење на циклуси во остаточниот транспортен систем ($G(V, E, c)$)

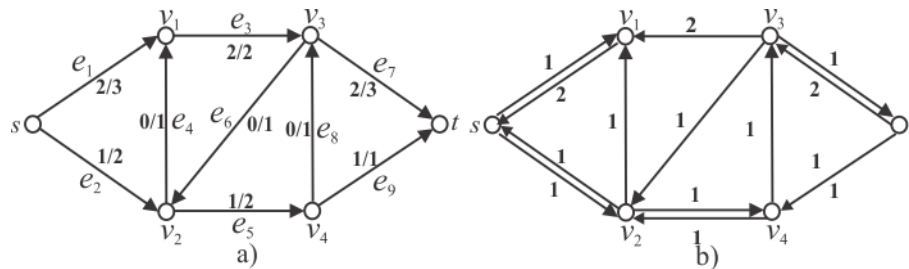
Чекор 1. Со користење на алгоритмот на Ford - Fulkerson се наоѓа еден проток f за ниво d .

Чекор 2. Додека $G(V, E^f)$ има циклус, се поставува $f = f_1$ со користење на (5.8).

Чекор 3. Се наоѓаат сите циклуси во остаточниот граф на f и се конструираат нивните зголемувачки вектори \vec{y}_k .

Чекор 4. Се проверува дали постои циклус во секој $\vec{x}' = \vec{x} + \sum_{k=1}^r \vec{y}_k$. Ако не постои циклус и ако за секој $i, x'_i \leq c(e_i)$ тогаш се печати \vec{x}' .

Пример 5.7 Го разгледуваме транспортниот систем од Фигура 5.7 која има 9 линкови и 6 јазли. Векторот од минимални капацитети е $(3, 2, 2, 1, 2, 1, 3, 1, 1)$ и ниво $d = 3$.



Фигура 5.7 а) Минимален пат вектор за ниво 3, $(2, 1, 2, 0, 1, 0, 2, 0, 1)$; б) остаточен транспортен систем индуциран од минималниот пат вектор $(2, 1, 2, 0, 1, 0, 2, 0, 1)$.

Чекор 1. Со користење на алгоритмот на Ford - Fulkerson се наоѓа минималниот пат вектор за ниво 3, $(2, 1, 2, 0, 1, 0, 2, 0, 1)$.

Чекор 2. Графот индуциран од $(2, 1, 2, 0, 1, 0, 2, 0, 1)$ нема циклуси.

Чекор 3. Се наоѓаат сите циклуси во остаточниот граф: $\mathcal{C}_1 = \langle s, v_2, v_1, s \rangle$, $\mathcal{C}_2 = \langle v_3, v_2, v_4, v_3 \rangle$, $\mathcal{C}_3 = \langle v_4, v_3, t, v_4 \rangle$ и $\mathcal{C}_4 = \langle s, v_2, v_4, v_3, v_1, s \rangle$. Соодветните зголемувачки вектори се: $\vec{y}_1 = (-1, 1, 0, 1, 0, 0, 0, 0, 0)$, $\vec{y}_2 = (0, 0, 0, 0, 1, 1, 0, 1, 0)$, $\vec{y}_3 = (0, 0, 0, 0, 0, 0, 1, 1, -1)$ и $\vec{y}_4 = (-1, 1, -1, 0, 1, 0, 0, 1, 0)$.

Чекор 4. Се проверува дали постои циклус во секој $\vec{x}' = \vec{x} + \sum_{k=1}^r \vec{y}_k$, $r = 1, 2, 3, 4$.

$\vec{x} + \vec{y}_1 = (1, 2, 2, 1, 1, 0, 2, 0, 1)$, графот индуциран од $\vec{x} + \vec{y}_1$ не содржи циклус од каде следува дека $\vec{x} + \vec{y}_1$ е 3-МР.

$\vec{x} + \vec{y}_2 = (2, 1, 2, 0, 2, 1, 2, 1, 1)$, графот индуциран од $\vec{x} + \vec{y}_2$ содржи циклус од каде следува дека $\vec{x} + \vec{y}_2$ не е 3-МР.

$\vec{x} + \vec{y}_3 = (2, 1, 2, 0, 1, 0, 3, 1, 0)$ графот индуциран од $\vec{x} + \vec{y}_3$ не содржи циклус од каде следува дека $\vec{x} + \vec{y}_3$ е 3-МР.

$\vec{x} + \vec{y}_4 = (1, 2, 1, 0, 2, 0, 2, 1, 1)$ графот индуциран од $\vec{x} + \vec{y}_4$ не содржи циклус од каде следува дека $\vec{x} + \vec{y}_4$ е 3- MP .

$\vec{x} + \vec{y}_1 + \vec{y}_2 = (1, 2, 2, 1, 2, 1, 2, 1, 1)$, графот индуциран од $\vec{x} + \vec{y}_1 + \vec{y}_2$ содржи циклус од каде следува дека $\vec{x} + \vec{y}_1 + \vec{y}_2$ не е 3- MP .

$\vec{x} + \vec{y}_1 + \vec{y}_3 = (1, 2, 2, 1, 1, 0, 3, 1, 0)$ графот индуциран од $\vec{x} + \vec{y}_1 + \vec{y}_3$ не содржи циклус од каде следува дека $\vec{x} + \vec{y}_1 + \vec{y}_3$ е 3- MP .

Истата постапка ја продолжуваме за сите комбинации од зголемувачки циклуси. Добиваме дека

$$3-MP = \{(2, 1, 2, 0, 1, 0, 2, 0, 1), (1, 2, 2, 1, 1, 0, 2, 0, 1), (1, 2, 1, 0, 2, 0, 2, 1, 1), (2, 1, 2, 0, 1, 0, 3, 1, 0), (1, 2, 2, 1, 1, 0, 3, 1, 0)\}.$$

Со овој алгоритам, наместо да се собираат d минимални пат вектори за ниво 1 за добивање на кандидати за d - MP како во алгоритамите предложени во [34], [36], [40] и [51], се врши само едно додавање на вектори за да се добие кандидат за d - MP . Овој пристап пред сè се користи за максималното ниво M , бидејќи во овој случај остаточниот граф може да се подели во неколку сврзани компоненти, така што секој циклус мора да лежи на една од овие компоненти.

Остаточниот граф добиен со користење на алгоритамот на Ford - Fulkerson за максимален проток M , може да се подели на строго сврзани компоненти, како што е објаснето во [16], [17]. Овие компоненти се користат за добивање на сите минимални пресеци. Линковите што лежат на некој минимален пресек мора да се користат со нивниот полн капацитет. Уште повеќе, секој циклус мора да лежи точно на една од овие компоненти. Со ова се намалува должината, како и бројот на циклуси. Исто така, секој M - MP вектор може да се добие со спојување на подвекторите кои одговараат на секоја строго поврзана компонента и подвекторите што ги поврзуваат строго сврзаните компоненти.

Нека $G(V, E, c)$ е двотерминален транспортен систем со максимален проток M и $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$ е остаточен транспортен систем за M - MP , \vec{x} . Да претпоставиме дека $G(V_k, E_k)$ се подграфови од $G(V, E)$, каде V_k е множество од сите јазли во k -тата поврзана компонента на $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$ и $E_k = \{(u, v) \mid u, v \in V_k\}$. Нека $E' = \bigcap_k \{(u, v) \notin E_k\}$, т.е. множеството од сите линкови што не се во сврзаните компоненти. Тогаш секој $e_i \in E'$ лежи на некој минимален пресек и $l_{\vec{x}}(e_i) = x_i$. Уште повеќе, за секој друг M - MP \vec{y} , $l_{\vec{y}}(e_i) = x_i$. Од друга страна, протокот што влегува во $G(V_k, E_k)$ е еднаков на протокот што излегува од $G(V_k, E_k)$, т.е. $l_{\vec{y}}(V \setminus V_k, V) = l_{\vec{y}}(V, V \setminus V_k) = d_k$.

Алгоритам 5.5 – Алгоритам за наоѓање на минимални пат вектори со користење на циклуси во остаточниот транспортен систем за максимално ниво M ($G(V, E, c)$)

Чекор 1. Со користење на алгоритмот на Ford - Fulkerson се наоѓа една функција на проток f за максималното ниво M .

Чекор 2. Се наоѓаат сите поврзани компоненти на остаточниот графот и се поставува $x_i = f(e_i)$ за сите $e_i \in E'$.

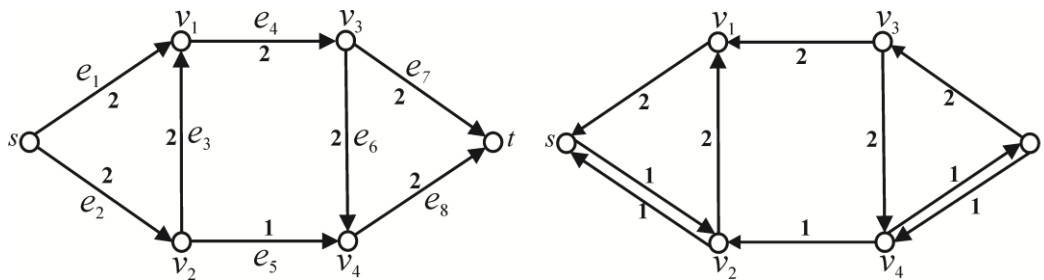
Чекор 3. За секоја строго поврзана компонента со користење на Алгоритам 5.4 се пресметува множеството од сите подвектори D_k .

Чекор 4. Се наоѓаат сите M -MPs \vec{y} за кои

$$y_i = \begin{cases} f(e_i), & e_i \in E' \\ l_{\vec{x}^k}(e_i), & e_i \in E_k \text{ и } \vec{x}^k \in D_k \end{cases} \quad (5.11)$$

Алгоритмот е илустриран со следниот пример:

Пример 5.8 Го разгледуваме транспортниот систем даден на Фигура 5.8 а), остаточниот транспортен систем за максималното ниво 3 за векторот индуциран од проток $(2, 1, 0, 2, 1, 0, 2, 1)$ е прикажана Фигура 5.8 б). Графот е поделен на два подграфови. Бидејќи компоненти во пресекот мора да се користат со нивниот полн капацитет, секој минимален 3-MP ја има формата $(x_1, x_2, x_3, 2, 1, x_6, x_7, x_8)$. Постои само еден зголемувачки циклус во првата строго поврзана компонента, кој ги содржи линковите e_1, e_2 и e_3 . Овој циклус е $(-1, 1, 1)$. Втората строго поврзана компонента ги содржи линковите e_6, e_7 и e_8 и исто така има еден зголемувачки циклус: $(1, -1, 1)$. Подвекторите кои одговараат на првата строго поврзана компонента за кои графот е ацикличен се $\{(2, 1, 0), (1, 2, 1)\}$, $((2 - 1, 1 + 1, 0 + 1) = (1, 2, 1))$, додека подвекторите кои одговараат на втората строго поврзана компонента се $\{(0, 2, 1), (1, 1, 2)\}$. Значи 3-MPs се векторите добиени со спојување на подвекторите од овие две множества, заедно со вредностите на компонентите на пресекот, т.е. 3-MP = $((2, 1, 0, 2, 1, 0, 2, 1), (1, 1, 2, 2, 1, 0, 2, 1), (2, 1, 0, 2, 1, 1, 1, 2), (1, 1, 2, 2, 1, 1, 1, 2))$.



Фигура 5.8 а) Проточен транспортен систем. б) Остаточен транспортен систем за векторот индуциран од проток $(2, 1, 0, 2, 1, 0, 2, 1)$.

Во продолжение ќе предложиме подобрен алгоритам на Алгоритам 5.4. Но пред да го предложиме алгоритмот ќе дадеме две теореми чии што резултати ги користиме во алгоритмот.

Со Теорема 5.5 е дадена врска помеѓу два d -MPs. Следнава теорема обезбедува посилен врска помеѓу два d -MPs споредено со Теорема 5.5.

Теорема 5.6 Нека \vec{x} и \vec{y} се два d -MPs, тогаш постојат прости циклуси $\mathcal{C}_i, i=1, \dots, k$

$$\vec{y} - \vec{x} = \sum_{i=1}^k \vec{y}^{\mathcal{C}_i,1}$$

Уште повеќе, $\forall i, j = 1, \dots, k$ и $\forall r = 1, \dots, |E|, y_r^{\mathcal{C}_i,1} \cdot y_r^{\mathcal{C}_j,1} \geq 0$, (т.е. не постојат јазли u и v така што (u, v) е во \mathcal{C}_i и (v, u) е во \mathcal{C}_j).

Доказ: Нека $\vec{z} = \sum_{i=1}^k \vec{y}^{\mathcal{C}_i,1}$. Тогаш $l_{\vec{z}}$ е проток со $|l_{\vec{z}}| = 0$. Затоа графот $G(V, E^{l_{\vec{z}}})$ има

прост циклус \mathcal{C}_1 . Сега ако и $\vec{z}_1 = \vec{z} - \vec{x}_{l_{\mathcal{C}_1}} \neq 0$, графот $G(V, E^{l_{\vec{z}_1}})$ има исто така прост циклус. Со продолжување на истата постапка ќе ги добиеме сите циклуси \mathcal{C}_i . Вториот дел од теоремата следи директно од фактот дека не постојат јазли u и v за кои и линкот (u, v) и линкот (v, u) се во $E^{l_{\vec{z}}}$. \square

Теорема 5.7 Нека \vec{x} и \vec{y} се два d -MPs, и $\mathcal{C}_i, i=1, \dots, k$ се прости циклуси така што

$$\vec{y} - \vec{x} = \sum_{i=1}^k \vec{y}^{\mathcal{C}_i,1}$$

Тогаш во секој циклус $\mathcal{C}_i, i=1, \dots, k$, постои линк (u, v) , за кој што линкот $(v, u) \in E^{l_{\vec{x}}}$ (т.е. $\exists r = 1, \dots, |E|, x_r \cdot y_r^{\mathcal{C}_i,1} < 0$).

Доказ: Без губење на општост можеме да претпоставиме дека во циклусот \mathcal{C}_1 не постои таков линк. Од Теорема 5.6 следува дека за секое $i = 2, \dots, k$ не постојат јазли u и v така што (u, v) е во \mathcal{C}_1 и (v, u) е во \mathcal{C}_i . Од овде следува дека векторите $\vec{z} = \vec{x} + \sum_{i=1}^k \vec{y}^{\mathcal{C}_i,1}$ и $\vec{y}^{\mathcal{C}_1,1}$ немаат координати со спротивни знаци. Од друга страна, јасно е дека векторот \vec{z} е вектор индуциран од проток за ниво d . како и векторот $\vec{z} + \vec{y}^{\mathcal{C}_1,1} = \vec{y}$. Но $\vec{z} < \vec{z} + \vec{y}^{\mathcal{C}_1,1} = \vec{y}$ што е контрадикција со нашата претпоставка дека \vec{y} е d -MP. \square

Сега можеме да го дадеме псевдокодот на подобрениот алгоритам.

Алгоритам 5.6 – Подобрен алгоритам за наоѓање на минимални пат вектори со користење на циклуси во остаточен транспортен систем $(G(V, E, c))$

Чекор 1. Се извршува алгоритмот на Ford-Fulkerson сè додека не се најде еден проток за ниво d . и соодветниот остаточен транспортен систем $G_f(V, E_f)$.

Чекор 1.1 Се додека во $G_f(V, E_f)$ постои циклус, се поставува $f = f_1$ со користење на 5.8.

Чекор 2. Се пресметува векторот \vec{x}_f и се поставува множеството $d\text{-}MP = \{ \vec{x}_f \}$.

Чекор 3. Се наоѓаат сите циклуси \mathcal{C}_i и за нив соодветните зголемувачки вектори $\vec{y}^{\mathcal{C}_i,1}$, во остаточниот транспортен систем $G_f(V, E_f)$, за кои $\exists r = 1, \dots, |E|, x_r \cdot y_r^{\mathcal{C}_i,1} < 0$.

Чекор 4. Се формира листа од зголемувачките вектори $\vec{y}^{\mathcal{C}_i,1}$ за циклусите што се добиваат во Чекор 3, $L = \{ \vec{y}^{\mathcal{C}_1,1}, \vec{y}^{\mathcal{C}_2,1}, \dots, \vec{y}^{\mathcal{C}_t,1} \}$.

Чекор 5. За секој зголемувачки вектор $\vec{y}^{\mathcal{C}_i,1} i = 1, 2, \dots, t$, се формира листата од индекси на циклусите кои може да му се додадат, $I_i = \{ j \geq i \mid \forall r = 1, \dots, |E|, y_r^{\mathcal{C}_i,1} \cdot y_r^{\mathcal{C}_j,1} \geq 0 \}$.

Чекор 6. Се дефинира $I_{\vec{x}_f} = \{ j \mid j = 1, \dots, t \}$, $candidates = \{ (\vec{x}_f, I_{\vec{x}_f}) \}$ и $allcand = \{ \vec{x}_f \}$.

Чекор 7. Се додека $candidates \neq \emptyset$ се извршуваат следниве чекори

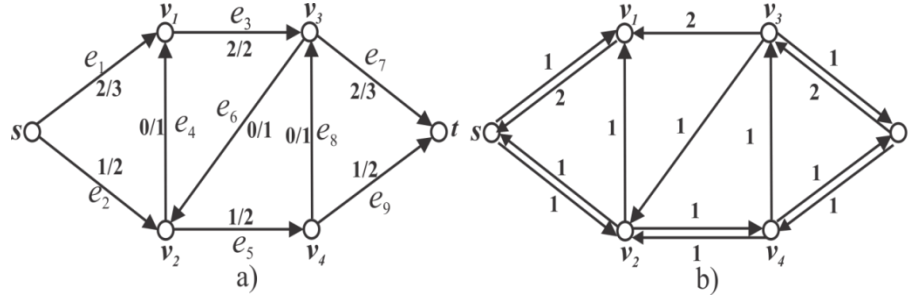
Чекор 7.1. Се зема првиот елемент од $candidates$ и се поставува \vec{y} да биде првата координата, $I_{\vec{y}}$ да биде втората координата од овој елемент и се брише елементот од $candidates$.

Чекор 7.2. За секој $j \in I_{\vec{y}}$ се формира векторот $\vec{z} = \vec{y} + \vec{y}^{\mathcal{C}_j,1}$ и ако не е во множеството $allcand$ се додава во ова множество. Ако $\vec{z} \in allcand$ се оди на Чекор 7.

Чекор 7.2.1 Ако $\forall r, z_r \leq c(e_r)$ и графот $G(V, E^{\vec{z}})$ индуциран од $I_{\vec{z}}$, нема циклуси се додава \vec{z} во $d\text{-}MP$.

Чекор 7.2.2 Ако $\forall r, z_r \leq c(e_r)$ се формира множеството $I_{\vec{z}} = I_{\vec{y}} + I_j$ и ако $I_{\vec{z}} \neq \emptyset$ се додава елементот $\{ (\vec{z}, I_{\vec{z}}) \}$ во $candidates$.

Пример 5.9 Да го разгледаме транспортниот систем прикажан на Фигура 5.9. Векторот од максимални капацитети е $(3, 2, 2, 1, 2, 1, 3, 1, 2)$. Ние треба да ги најдеме минималните пат вектори за ниво 3.



Фигура 5.9 Минимален пат вектор за ниво 3, $(2, 1, 2, 0, 1, 0, 2, 0, 1)$; b) остаточниот транспортен систем индуциран од минималниот пат вектор $(2, 1, 2, 0, 1, 0, 2, 0, 1)$.

Чекор 1. Еден минимален пат вектор за ниво 3 е $\vec{x}_f = (2, 1, 2, 0, 1, 0, 2, 0, 1)$.

Чекор 2. $3-MP = \{(2, 1, 2, 0, 1, 0, 2, 0, 1)\}$.

Чекор 3. Циклуси во транспортниот што имаат најмалку една координата со спротивен знак од соодветните координати на векторот се $\mathcal{C}_1 = \langle s, v_2, v_1, s \rangle$, $\mathcal{C}_2 = \langle s, v_2, v_4, v_3, v_1, s \rangle$, $\mathcal{C}_3 = \langle s, v_2, v_4, t, v_3, v_1, s \rangle$, $\mathcal{C}_4 = \langle v_2, v_4, t, v_3, v_2 \rangle$, $\mathcal{C}_5 = \langle v_3, t, v_4, v_3 \rangle$.

Чекор 4. $\vec{y}^{\mathcal{C}_1,1} = (-1, 1, 0, 1, 0, 0, 0, 0, 0)$, $\vec{y}^{\mathcal{C}_2,1} = (-1, 1, -1, 0, 1, 0, 0, 1, 0)$, $\vec{y}^{\mathcal{C}_3,1} = (-1, 1, -1, 0, 1, 0, -1, 0, 1)$, $\vec{y}^{\mathcal{C}_4,1} = (0, 0, 0, 0, 1, 1, -1, 0, 1)$, $\vec{y}^{\mathcal{C}_5,1} = (0, 0, 0, 0, 0, 0, 1, 1, -1)$.

$L = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5\}$

Чекор 5. $I_1 = \{1, 2, 3, 4, 5\}$, $I_2 = \{2, 3, 4, 5\}$, $I_3 = \{3, 4\}$, $I_4 = \{4\}$, $I_5 = \{5\}$.

Чекор 6. $I_{\vec{x}_f} = \{1, 2, 3, 4, 5\}$, $candidates = \{((2, 1, 2, 0, 1, 0, 2, 0, 1), \{1, 2, 3, 4, 5\})\}$ и $allcand = \{(2, 1, 2, 0, 1, 0, 2, 0, 1)\}$.

Чекор 7. Овој чекор е даден во табела. Во табелата се дадени векторите што се добиваат во секое поминување низ (“while”) јамката во Чекор 7. Во првата колона е прикажан i – от реден број на циклусот \mathcal{C}_i кој се додава на \vec{y} во овој чекор. Во втората колона е даден векторот \vec{z} , во третата колона пишуваме “True” кога \vec{z} се додава во множеството $allcand$, во спротивно пишуваме “False”. На сличен начин во четвртата колона пишуваме “True” кога \vec{z} е 3-MP, во спротивно пишуваме “False” и во последната колона ако \vec{z} се додава во множеството $allcand$ го пишуваме множеството $I_{\vec{z}}$.

Табелата 5.2 е за првиот елемент што се зема од множеството $candidates$, $\{((2, 1, 2, 0, 1, 0, 2, 0, 1), \{1, 2, 3, 4, 5\})\}$.

Табела 5.2. Минимални пат вектори за ниво 3 добиени од $\{(2, 1, 2, 0, 1, 0, 2, 0, 1), \{1, 2, 3, 4, 5\}\}$

$(\vec{y}, l_{\vec{y}}) = \{(2, 1, 2, 0, 1, 0, 2, 0, 1), \{1, 2, 3, 4, 5\}\}$				
e_i	\vec{z}	<i>allcand</i>	3-MP	$I_{\vec{z}}$
1	(1, 2, 2, 1, 1, 0, 2, 0, 1)	True	True	{1,2,3,4,5}
2	(1, 2, 1, 0, 2, 0, 2, 1, 1)	True	True	{2,3,4,5}
3	(1, 2, 1, 0, 2, 0, 1, 0, 2)	True	True	{3,4}
4	(2, 1, 2, 0, 2, 1, 1, 0, 2)	True	True	{4}
5	(1, 2, 1, 0, 1, 0, 3, 0, 1)	True	True	{5}

Табела 5.3 е за вториот елемент што се зема од *candidates*, $\{(2, 1, 2, 0, 1, 0, 2, 0, 1), \{1, 2, 3, 4, 5\}\}$.

Табела 5.3. 3-MPs добиени од $((1, 2, 2, 1, 1, 0, 2, 0, 1), \{1, 2, 3, 4, 5\})$.

$(\vec{y}, l_{\vec{y}}) = \{(1, 2, 2, 1, 1, 0, 2, 0, 1), \{1, 2, 3, 4, 5\}\}$				
e_i	\vec{z}	<i>allcand</i>	3-MP	$I_{\vec{z}}$
1	(0, 3, 2, 2, 0, 0, 2, 0, 1)	False	False	
2	(0, 3, 1, 1, 2, 0, 2, 1, 1)	False	False	
3	(0, 3, 1, 1, 2, 0, 1, 0, 2)	False	False	
4	(1, 2, 2, 1, 2, 1, 1, 0, 2)	True	True	{4}
5	(1, 2, 2, 1, 1, 0, 3, 1, 0)	True	True	{5}

Табела 5.4 е за следниот елемент што се зема од *candidates*, $\{(1, 2, 1, 0, 2, 0, 2, 1, 1), \{2, 3, 4, 5\}\}$.

Табела 5.4. 3-MPs добиени од $((1, 2, 1, 0, 2, 0, 2, 1, 1), \{2, 3, 4, 5\})$.

$(\vec{y}, l_{\vec{y}}) = \{(1, 2, 1, 0, 2, 0, 2, 1, 1), \{2, 3, 4, 5\}\}$				
e_i	\vec{z}	<i>allcand</i>	3-MP	$I_{\vec{z}}$
2	(0, 3, 0, 0, 3, 0, 2, 2, 1)	False	False	
3	(0, 3, 0, 0, 3, 0, 1, 1, 2)	False	False	
4	(1, 2, 1, 0, 3, 1, 1, 1, 2)	False	False	
5	(1, 2, 1, 0, 2, 0, 3, 2, 0)	False	False	

Табела 5.5 е за следните елементи што се земаат од *candidates*, $((1, 2, 1, 0, 2, 0, 1, 0, 2), \{3, 4\})$, $((2, 1, 2, 0, 2, 1, 1, 0, 2), \{4\})$ и $((1, 2, 1, 0, 1, 0, 3, 0, 1), \{5\})$.

Табела 5.5. 3-MPs добиени од $\{((1, 2, 1, 0, 2, 0, 1, 0, 2), \{3, 4\})\}$, $\{((2, 1, 2, 0, 2, 1, 1, 0, 2), \{4\})\}$ и $\{((1, 2, 1, 0, 1, 0, 3, 0, 1), \{5\})\}$.

$(\vec{y}, l_{\vec{y}}) = \{(1, 2, 1, 0, 2, 0, 1, 0, 2), \{3, 4\}\}$				
e_i	\vec{z}	<i>allcand</i>	3-MP	$I_{\vec{z}}$
3	(0, 3, 0, 0, 3, 0, 0, 0, 3)	False	False	
4	(1, 2, 1, 0, 3, 1, 0, 0, 3)	False	False	
$(\vec{y}, l_{\vec{y}}) = \{(2, 1, 2, 0, 2, 1, 1, 0, 2), \{4\}\}$				
4	(2, 1, 2, 0, 3, 2, 0, 0, 3)	False	False	
$(\vec{y}, l_{\vec{y}}) = \{(1, 2, 1, 0, 1, 0, 3, 0, 1), \{5\}\}$				
5	(1, 2, 1, 0, 2, 0, 4, 1, 0)	False	False	

Табела 5.6 е за следните елементи што се земаат од *candidates* $((1, 2, 2, 1, 2, 1, 1, 0, 2), \{4\})$ и $((1, 2, 2, 1, 1, 0, 3, 1, 0), \{5\})$.

Табела 5.6. 3-MPs добиени од $((1, 2, 2, 1, 2, 1, 1, 0, 2), \{4\})$ и $((1, 2, 2, 1, 1, 0, 3, 1, 0), \{5\})$.

$(\bar{y}, l_{\bar{y}}) = \{(1, 2, 2, 1, 2, 1, 1, 0, 2), \{4\}\}$				
e_i	\bar{z}	<i>allcand</i>	3-MP	$I_{\bar{z}}$
4	(1, 2, 2, 1, 3, 2, 0, 0, 3)	False	False	
$(\bar{y}, l_{\bar{y}}) = \{(1, 2, 2, 1, 1, 0, 3, 1, 0), \{5\}\}$				
5	(1, 2, 1, 0, 1, 0, 4, 2, -1)	False	False	

Во програмскиот јазик C# се направени програми за Алгоритмот 5.6 и за Алгоритмот 5.1 што го користи алгоритмот предложен од Lin et al. [51]. Извршивме тестирање на двата алгоритми за 10 графови со 6 јазли, 10 графови со 15 јазли и 6 графови со 30 јазли. Максималниот капацитет на транспортниот систем го зедеме да биде 10. Зедеме само 6 графови со 30 јазли, бидејќи со зголемување на бројот на линковите за поголемите нивоа (нивоа 8, 9 и 10) со Алгоритмот 5.1 се добиваат многу големи вредности, поточно беше потребно многу време да се чека за да се изврши програмата.

Во Табела 5.7 се дадени времињата што ги добивме со извршувањето на програмите. Во редиците е даден бројот на јазли во транспортниот систем, додека во колоните нивото на работа на системот (од 1 до 10). За секое ниво на работа и соодветен број на јазли се дадени времињата на извршување на програмите во милисекунди (ms), и за Алгоритмот 5.6 и за Алгоритмот 5.1.

Табела 5.7 Времиња на извршување на Алгоритам 5.6 и Алгоритам 5.1.

	d	1	2	3	4	5	6	7	8	9	10
Број на јазли											
Алгоритам 5.6	6	3	116,1	135,9	134,2	125,7	110,5	95,4	119,9	89,1	49,3
Алгоритам 5.1	6	3	32,7	76,11	365,6	958,6	2078,1	5105	11147,8	20817,8	31784,11
Алгоритам 5.6	15	3	243,6	566,7	1214,9	1771,3	1770,7	1005,4	672,8	342,7	173,3
Алгоритам 5.1	15	3	41,3	148,8	637,9	1988	4471,5	9748,5	20872,3	39880,11	52353,5
Алгоритам 5.6	30	3	601,5	895	1978,5	2189	2265	1742,5	954,17	704,17	181,3
Алгоритам 5.1	30	3	49,2	167,3	1231	1824,6	2210,1	2985,8	24295	54629	63104

Од Табела 5.7 можеме да забележиме дека за пониските нивоа (2 и 3) Алгоритмот 5.1 што го користи алгоритмот предложен во [51] е малку побрз од Алгоритам 5.6. Од друга страна за поголеми нивоа Алгоритам 5.6 е значително побрз од Алгоритам 5.1. Тоа се должи на фактот што за поголеми нивоа во остаточниот систем што е соодветен на минималниот пат вектор добиен со примена на алгоритмот на Ford-Fulkerson, некои од

линковите го достигнуваат максималниот капацитет на соодветниот линк, па со тоа и бројот на циклуси во остаточниот систем ќе биде помал.

На брзината на Алгоритмот 5.1 што го користи алгоритмот предложен во [51] најмногу влијае бројот на линковите во транспортниот систем и вредноста на нивото на работа. Поточно, со зголемување на бројот на линковите се зголемува и бројот на минимални патишта што значително влијае на брзината на извршување на Чекор 3 на алгоритмот на Lin, Jane и Yuan кој е разработен детално во Прилогот А. Уште повеќе брзината на извршување се зголемува со зголемувањето на нивото на работа, бидејќи во тој случај значително се зголемува бројот на можни решенија што ќе треба да проверуваме дали ги задоволуваат ограничувањата (A.1) – (A.3) дадени во Чекор 3 на алгоритмот на Lin, Jane и Yuan.

5.3 Подобрен алгоритам за наоѓање на минимални пат вектори за ненасочени транспортни системи

Во овој дел ќе дадеме подобрен алгоритам за наоѓање на минимални пат вектори за ниво d за ненасочени транспортни системи. Пред да го предложиме подобриениот алгоритам ќе изведеме некои заклучоци од теоремите дадени во Глава 4 и ќе докажеме едно својство, со чија помош ќе ја оптимизираме постапката за наоѓање на d -MPs.

Прво ќе дадеме една последица која директно следува од Теорема 4.4 и Теорема 4.5.

Последица 5.4 Даден е ненасочен двотерминален транспортен систем $G(V, E, c)$ со извор s , сливник t , граф од насоки $\hat{G}(V, \hat{E})$ и d -MP, \vec{x} . Тогаш постојат вектори индуцирани од проток $\vec{x}'_1, \dots, \vec{x}'_d$ со вредност 1, така што векторот \vec{x} е соодветен d -MP на векторот индуциран од проток со вредност d , $\vec{x}' = \vec{x}'_1, \dots, \vec{x}'_d$. Уште повеќе, за секои i и j , постои тополошко сортирање на јазлите од V кое може да биде истото тополошко сортирање за двата графови $G(V, E^{f_i})$ и $G(V, E^{f_j})$, каде f_i and f_j се протоци кои одговараат на \vec{x}'_i и \vec{x}'_j соодветно.

Овој резултат ни овозможува да ја оптимизираме постапката за наоѓање на кандидатите за d -MP. Всушност, ако ни е познато множеството од сите $(d - 1)$ -MPs, можеме да го најдеме множеството од сите d -MPs. Прво ги наоѓаме сите кандидати за минимални пат вектори за ниво d како сума на векторите индуцирани од проток што се соодветни на минималните пат вектори за ниво $d - 1$ и векторите индуцирани од проток што се соодветни на минималните пат вектори за ниво 1. Оптимизацијата се состои во тоа што наместо да ги наоѓаме сите овие суми, да ги најдеме само сумите од оние вектори кои индуцираат графови со исто тополошко сортирање. Ова воопшто не е лесно и оваа проверка ќе одземе исто време како и проверката за постоење на циклус во графот. Корисна стратегија што овде може да се примени е да не се собираат вектори кои имаат соодветни координати со спротивни знаци (+ и -), бидејќи овие вектори може да се добијат и на друг начин (Својство 5.3).

Својство 5.3 Секој кандидат за вектор индуциран од проток што е добиен со собирање на вектори кои имаат соодветни координати со спротивни знаци, може да се добие и на друг начин.

Доказ: Нека $\vec{x} \in Q'_1$ и $\vec{y} \in Q'_1$ се два вектори индуцирани од проток кои имаат една соодветна координата со спротивен знак. Претпоставуваме дека оваа координата кореспондира со линкот $\{u, v\}$. Тогаш постои пат \mathcal{P}_1 што ги поврзува s и u . Нека протокот $f_{\vec{x}}$ поминува од s до u низ пат \mathcal{P}_1 . Овој проток го користи линкот $\{u, v\}$ во насока од u кон v , и од v до сливникот t поминува низ патот \mathcal{P}_2 . Нека протокот $f_{\vec{y}}$ поминува од s до v низ патот \mathcal{P}'_1 , и го користи линкот $\{u, v\}$ во насока од v кон u , и потоа оди од u до сливникот t низ патот \mathcal{P}'_2 . Тогаш, $\vec{x} + \vec{y}$ може исто така да се добие и со собирање на векторите $\vec{x}' + \vec{y}'$, каде \vec{x}' е вектор од состојба што е соодветен на протокот што одговара на патот што минува низ патиштата \mathcal{P}_1 и \mathcal{P}'_2 и \vec{y}' е вектор од состојба што е соодветен на протокот што одговара на патот што минува низ патиштата \mathcal{P}_2 и \mathcal{P}'_1 .

Со користење на индукција, може лесно да се докаже дека тврдењето е точно за кои било два вектори индуцирани од проток $\vec{x} \in Q'_1$ и $\vec{y} \in Q'_1$ што имаат повеќе од една соодветна координата со спротивен знак.

Поопшто, да разгледаме два вектори $\vec{x} \in Q'_d$ и $\vec{y} \in Q'_1$ кои имаат соодветни координати со спротивен знак. Можеме да бидеме сигурни дека векторот $\vec{x} \in Q'_d$ може да се претстави како линеарна комбинација на векторите од Q'_1 . Од индуктивната претпоставка, секој од овие d вектори има еднаков знак со другите вектори на сите соодветни координати. (сумата $\vec{x} + \vec{y}$ претставува линеарна комбинација на $d + 1$ вектори од Q'_1). Ако \vec{y} има координата со спротивен знак од соодветна координата на \vec{x} , тогаш најмалку еден од d - те “собирачи” на \vec{x} (да го означиме овој собирок со \vec{x}_1) има координата со спротивен знак од соодветната координата на \vec{y} . Како што докажавме претходно, $\vec{x}_1 + \vec{y}$ може да се добие и на друг начин од каде следува дека и $\vec{x} + \vec{y}$ може да се добие на друг начин. \square

Да забележиме дека сè уште може да се добијат минимални пат вектори повеќе од еднаш. На пример векторот $(1, 1, -1, 2, 0, 0, 1, 1, 1)$ е вектор индуциран од проток за ниво 2 за ненасочениот двотерминален транспортен систем прикажан на Фигура 4.2 а) и граф од насоки прикажан на Фигура 4.2 б), кој одговара на 2-МР $(1, 1, 1, 2, 0, 0, 1, 1, 1)$. Овој вектор индуциран од проток може да се добие повеќе од еднаш: со собирање на векторите $(1, 0, 0, 1, 0, 0, 0, 1, 0)$ и $(0, 1, -1, 1, 0, 0, 1, 0, 1)$, и со собирање на векторите $(0, 1, -1, 1, 0, 0, 0, 1, 0)$ и $(1, 0, 0, 1, 0, 0, 1, 0, 1)$.

Сега можеме да го дадеме подобруениот алгоритам за наоѓање на множеството од сите минимални пат вектори за ниво d .

Алгоритам 5.7 – Подобрен алгоритам за наоѓање на минимални пат вектор на ненасочен транспортен систем $(G(V, E, c))$

Чекор 1. Се конструира графот $\tilde{G}(V, \tilde{E}, \tilde{c})$ и се конструира графот од насоки $\hat{G}(V, \hat{E})$.

Чекор 2. Се наоѓа множеството Q'_1 од сите вектори индуцирани од проток за ниво 1 и множеството Q_1 од сите 1-MPs за $\tilde{G}(V, \tilde{E}, \tilde{c})$.

Чекор 3. За секое d од 2 до максималниот капацитет на транспортниот систем се извршува

Чекор 3.1 За секој вектор во Q'_1 и секој вектор во Q'_{d-1} ако вектор немаат соодветни координати со спротивни знаци, се собираат векторите и се добива кандидатот \vec{x}' и ако за секој i , $i, x'_i \leq c(e_i)$ тогаш се извршуваат следниве чекори:

Чекор 3.1.1 Се наоѓа протокот f што е соодветен за \vec{x}' и се проверува дали постои циклус $G(V, E^f)$. Ако постои циклус кандидатот не се става во Q_d .

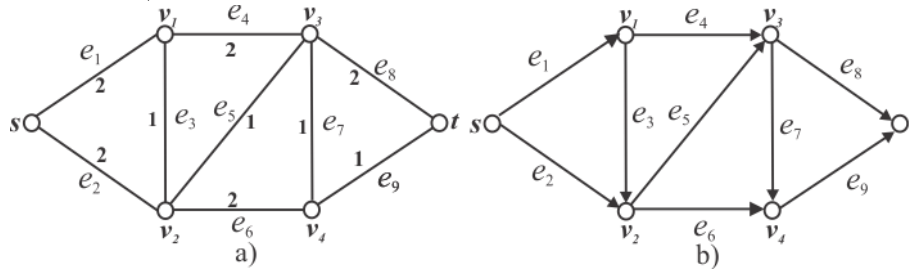
Чекор 3.1.2 Се проверува дали овој вектор е веќе во Q_d ако е во Q_d , кандидатот не се става во Q_d .

Временската комплексност на овој алгоритам е $O(|E|\lambda' + |E|\log(m))$, каде λ' е горната граница на бројот на добиени кандидати за сите нивоа и m е максималниот капацитет на линковите. Да забележиме дека алгоритамот ја има оваа комплексност за наоѓање на минималните пат вектори за сите можни протоци што може да бидат пренесени низ транспортниот систем.

Општо, овој пристап дава помал број на кандидати од алгоритамот на Lin, Jane и Yuan, уште повеќе помал е и бројот на кандидатите што се јавуваат повеќе од еднаш. Всушност, како што е покажано во Теорема 4.5, сите кандидати што се добиени со Алгоритамот 5.7 се линеарни комбинации од 1-MPs. Сите овие вектори ги задоволуваат ограничувањата (A.1) - (A.3) дадени во Прилог А ([51]), поради тоа што овие неравенки се наменети за проверување дали некој вектор користи капацитет што е поголем од максималниот капацитет. Поради ова, сите вектори што се добиваат со Алгоритамот 5.7 се добиваат и со алгоритамот предложен во [51]. Од друга страна, алгоритамот предложен во [51] ги и дава сите кандидати за d -MP што се добиваат од протоците што користат некој линк во спротивна насока. Алгоритамот 5.7 ги отстранува овие кандидати во Чекор 3.1. На пример, 3-MP векторот $(1, 2, 0, 1, 0, 2, 2, 1, 2)$ од Фигура 4.2 е добиен со алгоритамот на Lin, Jane и Yuan со собирање на протоците $(1, 0, 0, 1, 0, 1, 0, 0, 1)$, $(0, 1, 0, 0, 0, 1, 1, 1, 0)$ и $(0, 1, 0, 0, 0, 1, 0, 0, 1)$, додека тој вектор не се добива со Алгоритамот 5.7, бидејќи векторите што се индуцирани од првите два протоци, ќе имаат различен знак на седмата координата. Исто така, кандидатот f за кој графот $G(V, E^f)$ има циклус со капацитет поголем од 2 е добиен со алгоритамот предложен во [52], но не и со Алгоритамот 5.7. На пример, да го разгледаме транспортниот систем прикажан на Фигура 4.2, но земаме дека линкот e_5 има капацитет 2 и линкот e_6 има капацитет 4. Тогаш векторот $(2, 2, 0, 2, 2, 4, 2, 2, 2)$ ќе биде кандидат за 4-MP во согласност со алгоритамот во [51], бидејќи може да се добие како $2(1, 0, 0, 1, 1, 1, 0, 0, 1) + 2(0, 1, 0, 0, 0, 1, 1, 1, 0)$.

Со Алгоритамот 5.7 нема да се добие овој вектор, бидејќи единствен начин за да се добие овој вектор е со собирање на векторите индуцирани од проток $\vec{x} = (1, 0, 0, 1, -1, 1, 0, 0, 0, 1)$ и $\vec{y} = (0, 1, 0, 0, 0, 1, -1, 1, 0)$, што резултира во вектор индуциран од проток $2\vec{x} + 2\vec{y}$. Но не постои вектор индуциран од проток за ниво 3 кој кога ќе се собере со вектор од Q'_1 ќе го даде векторот $2\vec{x} + 2\vec{y}$, бидејќи ако постои вектор \vec{z} , така што $\vec{z} + \vec{z}' = 2\vec{x} + 2\vec{y}$, за некој вектор $\vec{z}' \in Q'_1$, тогаш $G(V, E^{\vec{z}})$ мора да содржи циклуси.

Пример 5.10 Да го разгледаме транспортниот систем што е прикажан на Фигура 5.10, кој има максимален капацитет 3.



Фигура 5.10 а) Ненасочен транспортен систем со вектор од максимален капацитет $(2, 2, 1, 2, 1, 2, 1, 2, 1)$; б) граф од насоки за ненасочениот транспортен систем.

$d = 1$: Множеството од вектори од проток за ниво 1, Q'_1 и множеството од сите 1-MPs, Q_1 се дадени во Табела 5.8.

Табела 5.8 Минимални пат вектори за ниво 1.

Q'_1	Q_1
$(1, 0, 1, 0, 1, 0, 1, 0, 1)$	$(1, 0, 1, 0, 1, 0, 1, 0, 1)$
$(1, 0, 1, 0, 1, 0, 0, 1, 0)$	$(1, 0, 1, 0, 1, 0, 1, 0, 1)$
$(1, 0, 1, 0, 0, 1, -1, 1, 0)$	$(1, 0, 1, 0, 0, 1, 1, 1, 0)$
$(1, 0, 1, 0, 0, 1, 0, 0, 1)$	$(1, 0, 1, 0, 0, 1, 0, 0, 1)$
$(1, 0, 0, 1, -1, 1, 0, 0, 1)$	$(1, 0, 0, 1, 1, 1, 0, 0, 1)$
$(1, 0, 0, 1, 0, 0, 1, 0, 1)$	$(1, 0, 0, 1, 0, 0, 1, 0, 1)$
$(1, 0, 0, 1, 0, 0, 0, 1, 0)$	$(1, 0, 0, 1, 0, 0, 0, 1, 0)$
$(0, 1, -1, 1, 0, 0, 1, 0, 1)$	$(0, 1, 1, 1, 0, 0, 1, 0, 1)$
$(0, 1, -1, 1, 0, 0, 0, 1, 0)$	$(0, 1, 1, 1, 0, 0, 0, 1, 0)$
$(0, 1, 0, 0, 1, 0, 1, 0, 1)$	$(0, 1, 0, 0, 1, 0, 1, 0, 1)$
$(0, 1, 0, 0, 1, 0, 0, 1, 0)$	$(0, 1, 0, 0, 1, 0, 0, 1, 0)$
$(0, 1, 0, 0, 0, 1, -1, 1, 0)$	$(0, 1, 0, 0, 0, 1, 1, 1, 0)$
$(0, 1, 0, 0, 0, 1, 0, 0, 1)$	$(0, 1, 0, 0, 0, 1, 0, 0, 1)$

$d=2$: Во Табела 5.9 се прикажани сите кандидати за вектори од проток Q'_2 и е објаснето како секој од нив е добиен. Ако кандидат за минимален пат вектор е минимален пат

вектор, тогаш е прикажан во колоната Q_2 , во спротивно во колоната Q_2 се содржи пораката “Постои циклус во $G(V, E^f)$ ” што значи дека кандидатот за минимален пат вектор не е минимален пат вектор. Кандидатите што се отфрлаат (тие не се минимални пат вектори) се $\bar{y}'_5, \bar{y}'_{10}$, и \bar{y}'_{11} .

Табела 5.9 Минимални пат вектори за ниво 2

\bar{y}'_i	Q'_2	Q_2
\bar{y}'_1	$\bar{x}'_2 + \bar{x}'_6 = (2, 0, 1, 1, 1, 0, 1, 1, 1)$	$(2, 0, 1, 1, 1, 0, 1, 1, 1)$
\bar{y}'_2	$\bar{x}'_2 + \bar{x}'_7 = (2, 0, 1, 1, 1, 0, 0, 2, 0)$	$(2, 0, 1, 1, 1, 0, 0, 2, 0)$
\bar{y}'_3	$\bar{x}'_2 + \bar{x}'_{12} = (1, 1, 1, 0, 1, 1, -1, 2, 0)$	$(1, 1, 1, 0, 1, 1, 1, 2, 0)$
\bar{y}'_4	$\bar{x}'_2 + \bar{x}'_{13} = (1, 1, 1, 0, 1, 1, 0, 1, 1)$	$(1, 1, 1, 0, 1, 1, 0, 1, 1)$
\bar{y}'_5	$\bar{x}'_3 + \bar{x}'_5 = (2, 0, 1, 1, -1, 2, -1, 1, 1)$	Постои циклус во $G(V, E^f)$
\bar{y}'_6	$\bar{x}'_3 + \bar{x}'_7 = (2, 0, 1, 1, 0, 1, -1, 2, 0)$	$(2, 0, 1, 1, 0, 1, 1, 2, 0)$
\bar{y}'_7	$\bar{x}'_3 + \bar{x}'_{13} = (1, 1, 1, 0, 0, 2, -1, 1, 1)$	$(1, 1, 1, 0, 0, 2, 1, 1, 1)$
\bar{y}'_8	$\bar{x}'_4 + \bar{x}'_7 = (2, 0, 1, 1, 0, 1, 0, 1, 1)$	$(2, 0, 1, 1, 0, 1, 0, 1, 1)$
\bar{y}'_9	$\bar{x}'_5 + \bar{x}'_7 = (2, 0, 0, 2, -1, 1, 0, 1, 1)$	$(2, 0, 0, 2, 1, 1, 0, 1, 1)$
\bar{y}'_{10}	$\bar{x}'_5 + \bar{x}'_9 = (1, 1, -1, 2, -1, 1, 0, 1, 1)$	Постои циклус во $G(V, E^f)$
\bar{y}'_{11}	$\bar{x}'_5 + \bar{x}'_{12} = (1, 1, 0, 1, -1, 2, -1, 1, 1)$	Постои циклус во $G(V, E^f)$
\bar{y}'_{12}	$\bar{x}'_6 + \bar{x}'_7 = (2, 0, 0, 2, 0, 0, 1, 1, 1)$	$(2, 0, 0, 2, 0, 0, 1, 1, 1)$
\bar{y}'_{13}	$\bar{x}'_6 + \bar{x}'_9 = (1, 1, -1, 2, 0, 0, 1, 1, 1)$	$(1, 1, 1, 2, 0, 0, 1, 1, 1)$
\bar{y}'_{14}	$\bar{x}'_6 + \bar{x}'_{11} = (1, 1, 0, 1, 1, 0, 1, 1, 1)$	$(1, 1, 0, 1, 1, 0, 1, 1, 1)$
\bar{y}'_{15}	$\bar{x}'_7 + \bar{x}'_7 = (2, 0, 0, 2, 0, 0, 0, 2, 0)$	$(2, 0, 0, 2, 0, 0, 0, 2, 0)$
\bar{y}'_{16}	$\bar{x}'_7 + \bar{x}'_9 = (1, 1, -1, 2, 0, 0, 0, 2, 0)$	$(1, 1, 1, 2, 0, 0, 0, 2, 0)$
\bar{y}'_{17}	$\bar{x}'_7 + \bar{x}'_{11} = (1, 1, 0, 1, 1, 0, 0, 2, 0)$	$(1, 1, 0, 1, 1, 0, 0, 2, 0)$
\bar{y}'_{18}	$\bar{x}'_7 + \bar{x}'_{12} = (1, 1, 0, 1, 0, 1, -1, 2, 0)$	$(1, 1, 0, 1, 0, 1, 1, 2, 0)$
\bar{y}'_{19}	$\bar{x}'_7 + \bar{x}'_{13} = (1, 1, 0, 1, 0, 1, 0, 1, 1)$	$(1, 1, 0, 1, 0, 1, 0, 1, 1)$
\bar{y}'_{20}	$\bar{x}'_8 + \bar{x}'_{11} = (0, 2, -1, 1, 1, 0, 1, 1, 1)$	$(0, 2, 1, 1, 1, 0, 1, 1, 1)$
\bar{y}'_{21}	$\bar{x}'_9 + \bar{x}'_{11} = (0, 2, -1, 1, 1, 0, 0, 2, 0)$	$(0, 2, 1, 1, 1, 0, 0, 2, 0)$
\bar{y}'_{22}	$\bar{x}'_9 + \bar{x}'_{12} = (0, 2, -1, 1, 0, 1, -1, 2, 0)$	$(0, 2, 1, 1, 0, 1, 1, 2, 0)$
\bar{y}'_{23}	$\bar{x}'_9 + \bar{x}'_{13} = (0, 2, -1, 1, 0, 1, 0, 1, 1)$	$(0, 2, 1, 1, 0, 1, 0, 1, 1)$
\bar{y}'_{24}	$\bar{x}'_{11} + \bar{x}'_{12} = (0, 2, 0, 0, 1, 1, -1, 2, 0)$	$(0, 2, 0, 0, 1, 1, 1, 2, 0)$
\bar{y}'_{25}	$\bar{x}'_{11} + \bar{x}'_{13} = (0, 2, 0, 0, 1, 1, 0, 1, 1)$	$(0, 2, 0, 0, 1, 1, 0, 1, 1)$
\bar{y}'_{26}	$\bar{x}'_{12} + \bar{x}'_{13} = (0, 2, 0, 0, 0, 2, -1, 1, 1)$	$(0, 2, 0, 0, 0, 2, 1, 1, 1)$

$d=3$: Во Табела 5.10 се прикажани сите кандидати за вектори од проток Q'_3 и е објаснето како секој од нив е добиен. Ако кандидат за минимален пат вектор е минимален пат вектор, тогаш е прикажан во колоната Q_3 , во спротивно во колоната Q_3 се содржи пораката “Постои циклус во $G(V, E^f)$ ” што значи дека кандидатот за минимален пат вектор не е минимален пат вектор. Кандидатите што се отфрлаат (тие не се минимални пат вектори) се \bar{z}'_4 и \bar{z}'_5 .

Табела 5.10 Минимални пат вектори за ниво 3.

\vec{z}'_i	Q'_3	Q_3
\vec{z}'_1	$\vec{x}'_2 + \vec{y}'_{19} = (2, 1, 1, 1, 1, 1, 0, 2, 1)$	$(2, 1, 1, 1, 1, 1, 0, 2, 1)$
\vec{z}'_2	$\vec{x}'_2 + \vec{y}'_{26} = (1, 2, 1, 0, 1, 2, -1, 2, 1)$	$(1, 2, 1, 0, 1, 2, 1, 2, 1)$
\vec{z}'_3	$\vec{x}'_3 + \vec{y}'_{19} = (2, 1, 1, 1, 0, 2, -1, 2, 1)$	$(2, 1, 1, 1, 0, 2, 1, 2, 1)$
\vec{z}'_4	$\vec{x}'_5 + \vec{y}'_{18} = (2, 1, 0, 2, -1, 2, -1, 2, 1)$	Постои циклус во $G(V, E')$
\vec{z}'_5	$\vec{x}'_3 + \vec{y}'_{22} = (1, 2, -1, 2, -1, 2, -1, 2, 1)$	Постои циклус во $G(V, E')$
\vec{z}'_6	$\vec{x}'_6 + \vec{y}'_{17} = (2, 1, 0, 2, 1, 0, 1, 2, 1)$	$(2, 1, 0, 2, 1, 0, 1, 2, 1)$
\vec{z}'_7	$\vec{x}'_6 + \vec{y}'_{21} = (1, 2, -1, 2, 1, 0, 1, 2, 1)$	$(1, 2, 1, 2, 1, 0, 1, 2, 1)$
\vec{z}'_8	$\vec{x}'_7 + \vec{y}'_{19} = (2, 1, 0, 2, 0, 1, 0, 2, 1)$	$(2, 1, 0, 2, 0, 1, 0, 2, 1)$
\vec{z}'_9	$\vec{x}'_7 + \vec{y}'_{23} = (1, 2, -1, 2, 0, 1, 0, 2, 1)$	$(1, 2, 1, 2, 0, 1, 0, 2, 1)$
\vec{z}'_{10}	$\vec{x}'_7 + \vec{y}'_{25} = (1, 2, 0, 1, 1, 1, 0, 2, 1)$	$(1, 2, 0, 1, 1, 1, 0, 2, 1)$
\vec{z}'_{11}	$\vec{x}'_7 + \vec{y}'_{26} = (1, 2, 0, 1, 0, 2, -1, 2, 1)$	$(1, 2, 0, 1, 0, 2, 1, 2, 1)$

6.Заклучок

Цел на проучувањето на докторската дисертација се двотерминалните повеќе-состојбени транспортни системи, поточно наоѓање на методи и техники за пресметување на нивната надежност. Во литературата постојат голем број на техники за пресметување на надежноста, кои може да се класифицираат во две главни категории, со првите надежноста се пресметува директно, додека со вторите прво се набројуваат сите можни поврзувања помеѓу јазлите и потоа се оценува надежноста. Ние се определивме за вториот пристап. Поточно во докторската дисертација предложивме нови алгоритми за наоѓање на минимални пат вектори за двотерминални повеќе - состојбени транспортни системи. При разгледувањето на познатите алгоритми во литературата и нивното анализирање, ние дојдовме до интуитивен заклучок дека графот што е индуциран од минимален пат вектор е ацикличен. Исто така заклучивме дека не постои доказ за нашиот “интуитивен заклучок” дека графот што е добиен од линковите на еден минимален пат вектор е ацикличен. Токму од овде произлезе и главната идеја за развој на новите алгоритми.

Во докторската дисертација (Глава 4) предложуваме теорија што ја поврзува теоријата на повеќе-состојбената надежност со теоријата на проток во двотерминални транспортни системи. Да забележаме дека во тековната литература за повеќе-состојбена надежност не најдовме слични истражувања од овој тип. Во дисертацијата покажавме дека протокот за одредено ниво ќе соодветствува со минимален пат вектор за истото ниво ако и само ако графот што е индуциран од тој проток е ацикличен. Ова тврдење го покажавме и за насочени и за ненасочени двотерминални транспортни системи. Исто така тоа ни помогна да дизајнираме алгоритми за наоѓање на минимални пат вектори, што се многу поефикасни од познатите алгоритми во литературата.

Првиот алгоритам што го предложивме во Глава 5 ги наоѓа минималните пат вектори и за насочени и за ненасочени транспортни системи. Земајќи во предвид дека минималните пат вектори индуцираат ацикличен граф, се дефинира релација на достапност во ацикличниот граф која дефинира подредување на јазлите во графот. Оваа релација на достапно е транзитивен затворач на релацијата на соседство. Бидејќи

процедурата за наоѓање транзитивен затворац, во општ случај има голема сложеност, ние покажавме дека транзитивниот затворац може да се добие со три множења на матрицата на соседство. Врз основа на оваа матрица на достапно утврдуваме дали графот што го формира некој минимален пат вектор е ацикличен или не.

Во докторската дисертација предложивме алгоритам кој работи само за насочени транспортни системи. Основната верзија на алгоритамот ја користи Теорема 4.1 за проверка дали некој кандидат за минимален пат вектор е навистина минимален. Во продолжение дадовме нова теорија со која се дава врската помеѓу два минимални пат вектори, и токму на оваа идеја се базира еден од подобрените алгоритми за насочени транспортни системи. Идејата ја развој на оваа теорија доаѓа од тоа што со користење на добро познатиот алгоритам на Ford-Fulkerson ние можеме да најдеме една проток за максимално ниво. Со негова адаптација можеме да најдеме и проток за одредено ниво и минималниот пат вектор што е соодветен за тој проток. Со користење на новата теорија што ја дава врската помеѓу минималните пат вектори ние ги наоѓаме сите минимални пат вектори. Да нагласиме дека нашите алгоритми ќе дадат најдобри резултати кога се бараат минималните пат вектори за максимално ниво.

Исто така се предложени и алгоритми за наоѓање на минимални пат вектори во ненасочен двотерминален повеќе - состојбен проточен транспортен систем. Во литературата најголем број од алгоритмите за наоѓање на минимални пат вектори се наменети за насочени мрежи. Од друга страна пак, постојат голем број на системи како телекомуникациски системи, транспортни системи, компјутерски мрежи и други кои е природно да се моделираат со ненасочени мрежи. Од овие причини, во докторската дисертација ние предлагаме и алгоритми за наоѓање на минимални пат вектори за овој тип на системи. Дадени се два алгоритми, за развој на првиот алгоритам се користат тврдењата дадени со Теорема 4.2 и Теорема 4.3. Додека вториот алгоритам е подобрување на првиот. Во овој алгоритам го искористивме својството што е и докажано во дисертацијата дека секој минимален пат вектор што се добива со собирање на вектори кои имаат соодветни координати со спротивни знаци може да се добие и на друг начин, без користење на вакви вектори. Па во овој алгоритам не ги земаме предвид векторите кои имаат соодветни координати со спротивни знаци.

Користена литература

- [1] A.Bethel, A. Gebre, Jose E. Ramirez-Marquez, "Element Substitution Algorithm for General Two-terminal Network Reliability Analyses," *IIE Transactions*, Volume 39, pp. 265 – 275, 2007.
- [2] A.Balakrishnan, "Models for Planning the Evolution of Local Telecommunication Networks", *Alfred P. Sloan School of Management, Massachusetts Institute of Technology*, 1989.
- [3] A.Kamil, "Graph Algorithms," *CS61B*, University of California, Berkeley, 2003.
- [4] A.Lisnianski, G.Levitin, "Multi-State system Reliability," *The Israel Electric Corporation Limited*, Israel, 2003.
- [5] A. Marteli, "A Gaussian Elimination Algorithm for the Enumeration of Cut Sets in a Graph," *Journal of the Association for Computing Machinery*, vol 23, no 1, 1976.
- [6] A.Wilson, N.Limnios, S.Keller Mc-Nulty, Y.Armijo, "Modern Statistical and Mathematical Methods in Reliability," *World Scientific Publishing*, 2005.
- [7] B.Natvig, S.Sørmo, A.Holen and G. Høgåsen, "Multistate Reliability Theory – A Case Study," *Advances in Applied Probability*, vol 18, pp. 921-932, 1986.
- [8] C.C. Jane, J.S. Lin, and J. Yuan, "Reliability evaluation of a limited-flow network in terms of minimal cut sets," *IEEE Transactions on Reliability*, vol. 42, pp. 354-361, 1993.
- [9] C. J. Colbourn, "Some open problems on reliability polynomials," *Technical Report 93-28*, University of Waterloo, Canada, 1993.
- [10] C. J. Colbourn, "The Combinatorics of Network Reliability," *Oxford University Press, Inc.*, New York, NY, USA, 1987.
- [11] C. Lucet and J. Manouvrier, "Exact methods to compute network reliability," *In MMR'97, 1st International Conference on Mathematical Methods in Reliability*, Bucharest, Romania, 1997.
- [12] D. Torrieri, "Calculation of Node-Pair Reliability in Large Networks with Unreliable Node," *IEEE Transactions on Reliability*, Vol 43, n 3, pp. 375-379, 1994.
- [13] E.Hansler, "A Fast Recursive Algorithm to Calculate the Reliability of a Communication Network," *IEEE Transactions on Communications*, Vol 20, n 3, pp. 637-640, 1972.
- [14] H. Salehi Fathabadi, M. Forghani-elahabadi, "A simple approach to search for all d-MCs of a limited-flow network," *Reliability Engineering and System Safety* 94, pp. 1878-1880, 2009.
- [15] H.T. Cormen, E.C. Leiserson, R.L. Rivest, C. Stein, "Introduction to Algorithms", 3rd edition, *The MIT Press*, 2009.
- [16] I. Bezakova, A.J. Friedlander, "Counting Minimum (s, t)-Cuts in Weighted Planar Graphs in Polynomial Time", In: *Proceedings of 35th International Symposium, MFCS 2010*, Brno, Czech Republic, August 23–27, 2010.
- [17] J.C. Picard, Q.Maurice, "On the structure of all minimum cuts in a network and application," *Mathematical Programming Study*, 13, pp. 8–16, 1980.
- [18] J.Erickson "Algorithms", University of Illinois at Urbana-Champaign, 2009.
- [19] J.E. Ramirez-Marquez and D.W. Coit, "Alternative Approach for Analyzing Multi-state Network Reliability," *Proceedings of the Industrial Engineering Research Conference (IERC)*, Portland, OR, May 2003.
- [20] J.E. Ramirez-Marquez and D.W. Coit, "Composite Importance Measures for Multi-State Systems with Multi-state Components," *IEEE Transactions on Reliability*, Vol 54, No3, September 2005.
- [21] J.E. Ramirez-Marquez and D.W. Coit, "Multi-state Component Criticality Analysis for Reliability Improvement in Multi-state Systems," *Reliability Engineering & System Safety*, Vol. 92, No. 10, October 2007.
- [22] J.E Ramirez-Marquez, D.W. Coit and M.Tortorella, "Multi-state Two-terminal Reliability: A Generalized Cut-Set Approach," *Rutgers University IE*, Working Paper, 2003.
- [23] J. Graver and M. Sobel, "You may rely on the reliability polynomial for much more than you might think," *Communications in Statistics- Theory and Methods*, pp. 1411–1422. 34(6), 2005.
- [24] J.Schroeder, P.A. Guedes, P. D.Jr., "Elias Computing the Minimum Cut and Maximum Flow of Undirected Graphs," *Relatório Técnico RT-DINF 003/2004*, Curitiba, PR, 2004.
- [25] J.S. Provan, M.O. Balls, "The complexity of counting cuts and of computing the probability that a graph is connected," *SIAM Journal on Computing*, 12(4), pp.777–788, 1983.

- [26] K.Aggarwal, "Integration of Reliability and Capacity in Performance Measure of a Telecommunication Network," *IEEE Transactions on Reliability*, Vol 34, n 2, pp. 184-186, 1985.
- [27] K.Aggarwal, Y.Chopra and J.Bajwa, "Capacity Consideration in Reliability Analysis of Communication Systems," *IEEE Transactions on Reliability*, Vol 31, n 2, 1982. pp. 177-180.
- [28] L.R. JR. Ford, D.E. Fulkerson, "Flows in Networks," Princeton University Press, 1962.
- [29] Martin. L. Shooman, "Reliability of computer systems and networks," *John Wiley and Sons*, 2002.
- [30] Ming J. Zuo, Way Kuo, "Optimal Reliability Modeling: Principles and Applications," *John Wiley and Sons*, 2002.
- [31] Min Xie, Yuan-Shun Dai, Kim-Leng Poh, "Computing System Reliability," *Kluwer Academic Publishers*, 2004.
- [32] M. J. Zuo, Z. Tian and HZ Huang "An efficient method for reliability evaluation of multistate networks given all minimal path vectors" - *IIE Transactions*, 39, pp.811-817, 2007.
- [33] M.J.Crowder, A.C. Kimber, R.L. Smith, T.J. Sweeting, "Statistical Analysis of Reliability Data," *Chapman and Hall*, London, 1991.
- [34] M. Forghani-elahabad, N. Mahdavi-Amiri, N "A simple algorithm to find all minimal path vectors to demand level d in a stochastic-flow network," *5-th Iranian Conference on Applied Mathematics*, Sep., 2013 Bu-Ali Sina University.
- [35] M. K. Chari and C. J. Colbourn, "Reliability polynomials: a survey", *Journal of Combinatorics, Information and System Sciences*, pp. 177-193. 22, 1998.
- [36] M. Mihova, N. Synagina, "An algorithm for calculating multi-state network reliability using minimal path vectors", *The 6th international conference for Informatics and In-formation Technology (CIIT 2008)* pp. 12-16, 2008.
- [37] M. Mihova, N. Maksimova, "Minimal Cut Sets for Transportation System", *Proceedings of the 7th International Conference for Informatics and Information Technology (CIIT 2010)*, ISBN 978-9989-668-88-3, pp.188-193, 2010.
- [38] M.Mihova, N. Maksimova, KJ. Gjorgjiev, "Optimal improving of network reliability: Reliability of the improved network", *Proceedings of the IV Congress of the Mathematicians of Republic of Macedonia*, Skopje, pp. 247-257, 2010.
- [39] M.Mihova, N. Maksimova, "Analysis of an Algorithm for finding Minimal Cut Set", *ICT Innovations 2010 Web Proceedings* ISSN 1857-7288, pp.299-308, 2010.
- [40] M.Mihova and N. Maksimova, N, "Estimation of minimal path vectors of multi state two terminal networks with cycles control", *Mathematica Balkanica*, Vol. 25, Fasc4, ISSN 0205-3217, pp. 437-447, 2011.
- [41] M. Mihova, N. Stojkovikj, M.Jovanov, E.Stankov, "On maximal level minimal path vectors of a two-terminal network" *International Journal of Olympiads in Informatics, IOI ,2014*, Vol. 8, pp. 133 - 144, 2014.
- [42] M. Mihova, N. Stojkovikj, M.Jovanov, E.Stankov, "On Maximal Level Minimal Path Vectors of a Two-terminal Undirected Network", *IEEE Transactions on Reliability*, (accepted paper).
- [43] M.Михова, "Моделирање на надежност на непоправливи повеќе-состојбени системи со независни компоненти и хомогени маркови транзиции," докторска дисертација, *Природно-математички факултет*, Скопје 2008.
- [44] M.Михова, "Моделирање на надежност на системи со повеќе состојби," магистерски труд, *Природно-математички факултет*, Скопје 2004.
- [45] M.Mihova, N.Maksimova, Z.Popeska, "An algorithm for calculating multi-state network reliability with arbitrary capacities of the links," - *Fourth International Bulgarian-Greek Conference Computer Science*, pp.70-75, 2008.
- [46] Н.Максимова "Надежност на повеќе-состојбени двотерминални транспортни системи," магистерски труд, *Природно-математички факултет*, Скопје 2009.
- [47] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, "Network Flows Theory, Algorithms, and Applications," *New Jersey: Englewood Cliffs, Prentice-Hall International*, 1993.
- [48] R. Terruggia, "Reliability analysis of probabilistic network," Ph.D. dissertation, University of Turin, 2010.
- [49] S. Kuo, S.Lu and F.Yeh, "Determining Terminal Pair Reliability Based on Edge Expansion Diagrams Using OBDD," *IEEE Transactions on Reliability*, Vol 48, n 3, pp. 234-246, 1999.
- [50] S.Lee, "Reliability Evaluation of a Flow network," *IEEE Transactions on Reliability*, Vol 29, n1, pp. 24-26, 1980.

- [51] S. J. Lin, C. C. Jane, J. Yuan, "On reliability evaluation of a capacitated – flow network in terms of minimal path sets," *Networks*, 3, pp. 131–138, 1995.
- [52] S. Patra and B. Misra, "Evaluation of Probability Mass Function of Flow in a Communication Network Considering a Multi-state Model of Network Links," *Microelectronics and Reliability*, Vol 36, n 3, pp. 415-421, 1996.
- [53] S. Patra, S. and B. Misra, "Reliability Evaluation of Flow Networks Considering Multi-state Modeling of Network Elements," *Microelectronics and Reliability*, Vol 33, n 14, pp. 2161-2164, 1993.
- [54] W. Dotson and J. Gobien, "A New Analysis Technique for Probabilistic Graphs," *IEEE Transactions on Circuits and Systems*, Vol 26, n 10, pp. 855-865, 1979.
- [55] WC. Yeh, "A simple MC-based Algorithm for Evaluating Reliability of a Stochastic-Flow Using Minimal Cuts to Evaluate the System Reliability of a Stochastic-Flow Network with Unreliable Nodes," *Reliability Engineering & System Safety*, 2003.
- [56] WC. Yeh, "A simple approach to search for all d-MCs of a limited-flow network" *Reliability Engineering and System Safety*, 71(1):15–9. 2001.
- [57] WC Yeh, "A Fast Algorithm for Searching All Multi-State Minimal Cuts," *IEEE Transaction on reliability*, vol. 57, no. 4, 2008.
- [58] WC Yeh, "A new approach to the d-MC problem," *Reliab Eng Syst Safety*, vol 77, issue 2, pp. 201–206, 2002.
- [59] Y. Lin, "Using Minimal Cuts to Evaluate the System Reliability of a Stochastic-Flow Network with Failures at Nodes and Arcs," *Reliability Engineering & System Safety*, Vol 75, pp. 41-46, 2002.
- [60] Y. K. Lin, "A simple algorithm for reliability evaluation of a stochastic-flow network with node failure," *Computers and Operations Research*, vol. 28, pp. 1277–1285, 2001.
- [61] Y. Yoo and N. Deo, "A Comparison of Algorithms for Terminal Pair Reliability," *IEEE Transactions on Reliability*, Vol 37, n 2, pp. 210-215, 1988.
- [62] Yi-Kuei Lin, Cheng-Ta Yeh, "Maximal network reliability with optimal transmission line assignment for stochastic electric power networks via genetic algorithms," *Applied Soft Computing* 11, pp. 2714–2724, 2011.
- [63] Z. Yan, M. Qian, "Improving efficiency of solving d-MC problem in stochastic-flow network," *Reliability Engineering & System Safety*, Volume 92, Issue 1, pp. 30–39, 2007.
- [64] YK Lin and HC Chou "System Reliability Evaluation of Data Transmission in Commercial Banks with Multiple Branches," *Hindawi Publishing Corporation Mathematical Problems in Engineering* Volume 2014, Article ID 584684, 8 pages, 2014.

Прилог А

Алгоритми за наоѓање на минимални пат вектори предложени од Y.S. Lin, C.C Jane и J. Yuan и M. Forghani – elahabad и N. Mahdavi-Amiri

Во овој прилог прво ќе го разгледаме алгоритмот за добивање на минимални пат вектори предложен во [51]. Потоа ќе разгледаме алгоритам што претставува модификација на првиот алгоритам, кој е предложен во [34].

Се разгледува стохастичен насочен проточен транспортен систем $G(V, E, c)$ со множество од јазли $V = \{v_1, v_2, \dots, v_{|V|}\}$, множество од линкови $E = \{e_1, e_2, \dots, e_{|E|}\}$ и функција од капацитет c . Да претпоставиме дека i - от линк може да работи во едно од нивоата од множеството $\{0, 1, \dots, M_i\}$, додека максималниот проток што може да се пренесе низ транспортниот систем да го означеме со M . ($\{0, 1, \dots, M\}$ е множество од сите можни протоци).

Авторите претпоставуваат дека $|E| > |V|$ па може да се земе $|E| = |V| + k$. Исто така претпоставуваат дека транспортниот систем G ги задоволува следниве претпоставки.

- Капацитетот на секој линк e_i е целобројна случајна променлива која може да прими една од вредностите $0 < 1 < 2 < \dots < M_i$ во согласност со дадената распределба.
- Протокот во G мора да ги задоволува законите за одржување на протокот (дадени во Глава 3, закон за ограничување на капацитетот и закон за одржување на проток.).
- Капацитетите на различни линкови се статистички независни.

Нека $\vec{x} = (x_1, x_2, \dots, x_{|E|})$ е вектор од состојби на системот (каде x_i е моменталниот капацитет на линкот e_i кога транспортниот систем е во состојба \vec{x} и може да прими една од вредностите $0 < 1 < 2 < \dots < c(e_i)$). За секој вектор од состојби \vec{x} , $\varphi(\vec{x})$ е возможниот капацитет од изворот до сливникот, ако системот е во состојба \vec{x} . Всушност, функцијата φ е структурната функција што ја дефинираме во Глава 2.

Нека системот е во состојба \vec{x} , множеството од линкови E ги има следниве три значајни подмножества:

$$N_{\vec{x}} = \{e_i \in E \mid x_i > 0\}, Z_{\vec{x}} = \{e_i \in E \mid x_i = 0\} \text{ и } S_{\vec{x}} = \{e_i \in N_{\vec{x}} \mid \varphi(\vec{x} - \vec{a}_i) < \varphi(\vec{x})\},$$

Каде $\vec{a}_i = (\delta_{i1}, \delta_{i2}, \dots, \delta_{i|E|})$, со $\delta_{ij} = 1$ ако $i = j$ и $\delta_{ij} = 0$ ако $i \neq j$. Всушност $E = S_{\vec{x}} \cup (N_{\vec{x}} / S_{\vec{x}}) \cup Z_{\vec{x}}$.

Векторот на состојби \vec{x} е минимален пат вектор со ниво d ако $\varphi(\vec{x}) = d$ и $N_{\vec{x}} = Z_{\vec{x}}$. Ако е потребно системот да работи со ниво d тогаш надежноса за ниво d $M2TR_d = P(\varphi(\vec{x}) \geq d)$ е веројатноста дека поток поголем или еднаков на d , може да биде успешно пренесен од изворот до сливникот.

Во следниот дел да го претставиме моделот, според кој авторите го конструираат својот алгоритам.

Да претпоставиме дека $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_p$ се минимални патишта на системот од s до t , од нив може да се најдат минимални пат вектори за ниво 1. Тоа се всушност минималните пат вектори за бинарниот транспортен систем кој ја има истата структура како и повеќе - состојбниот транспортен систем, $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_p$. Множеството од минимални пат вектори за ниво 1 ќе го означиме со $1-MP = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_p\}$.

Со овој алгоритам, минималните пат вектори за ниво d се добиваат со собирање на минималните пат вектори за ниво 1. Меѓутоа, авторите воведуваат дополнителни услови со чија помош се даваат одредени ограничувања. Овие ограничувања се однесуваат на собирањето на минималните пат вектори за ниво 1. Поточно се воведува индикатор со чија помош се одредува колку пати најмногу може некој минимален пат вектор за ниво 1 да се појави за добивање на минималните пат вектори за ниво d , и индикатор кој покажува кои минимални пат вектори за ниво 1 и колку пати некој од нив се користи при добивање на минимален пат вектор за ниво d . Подолу подетално ќе ги објаснеме овие индикатори.

За секој $\vec{x}_j \in 1-MP$ земаме индикатор $c_{\vec{x}_j} = \min\{M_i \mid x_{ji} = 1\}$. За да се добијат сите минимални пат вектори за ниво d , се конструираат сите d множества од $1-MP$, така што секој вектор $\vec{x}_j \in 1-MP$ се појавува најмногу $c_{\vec{x}_j}$ пати.

Другиот индикатор ќе го означиме со $c_{\vec{f}} = (c_{f_1}, c_{f_2}, \dots, c_{f_p})$, каде c_{f_j} покажува колку пати j - от минимален вектор за ниво 1 се собира при добивање на некој минимален пат вектор за ниво d , $\vec{y} = (y_1, y_2, \dots, y_{|E|})$. Овие индикатори ги задоволуваат следните услови:

$$\sum_{j=1}^p c_{f_j} = d. \quad (A.1)$$

$$c_{f_j} \geq c_{\vec{x}_j}, \quad j = 1, 2, \dots, p. \quad (A.2)$$

$$\sum_j \{c_{f_j} \mid x_{f_j} = 1\} \leq M_i, \quad i = 1, 2, \dots, |E|. \quad (\text{A.3})$$

Да забележиме дека $\sum_j \{c_{f_j} \mid x_{f_j} = 1\}$ е вкупниот проток кој поминува низ линкот e_i , под претпоставка дека $c_{\vec{f}} = (c_{f_1}, c_{f_2}, \dots, c_{f_p})$, векторот од состојби е \vec{y} . Всушност $\sum_j \{c_{f_j} \mid x_{f_j} = 1\}$ не го надминува максималниот проток M_i на линкот e_i .

Ова е дадено во следнава лема:

Лема А.1 Нека $\vec{y} = (y_1, y_2, \dots, y_{|E|})$ е вектор од состојба на системот за кој важи $\varphi(\vec{y}) = d$. Тогаш следниот услов е потребен за да се зачува законот за одржување на проток за векторот од состојби на системот \vec{y} :

$$y_i \geq \sum_j \{c_{f_j} \mid x_{f_j} = 1\} \leq M_i, \quad i = 1, 2, \dots, |E|. \quad (\text{A.4})$$

за кој било вектор $c_{\vec{f}} = (c_{f_1}, c_{f_2}, \dots, c_{f_p})$ за кој важи $\sum_{j=1}^p c_{f_j} = d$.

Лема А.2 Нека \vec{y} е минимален пат вектор за ниво d . Тогаш следниот услов е потребен за да се зачува законот за одржување на проток за векторот на состојби на системот \vec{y} :

$$y_i = \sum_j \{c_{f_j} \mid x_{f_j} = 1\} \leq M_i, \quad i = 1, 2, \dots, |E|. \quad (\text{A.5})$$

за кој било вектор $c_{\vec{f}} = (c_{f_1}, c_{f_2}, \dots, c_{f_p})$ за кој важи $\sum_{j=1}^p c_{f_j} = d$.

Кој било вектор $\vec{y} = (y_1, y_2, \dots, y_{|E|})$, кој е добиен со примена на (A.1) - (A.3) и потоа трансформиран во согласност со равенката (A.5) е кандидат за минимален пат вектор за ниво d . Секој минимален пат вектор може да биде добиен од кандидатите за минимални пат вектори. Со предложениот алгоритам, прво со методот на имплицитно набројување, се наоѓаат решенија кои ги задоволуваат ограничувањата (A.1) - (A.3) и потоа, со трансформација, овие целобројни решенија се трансформираат во кандидати за минимални пат вектори за ниво d со помош на равенството (A.5). Всушност равенството A.5 може да се напише и на друг начин, т.е ако $c_{\vec{f}} = (c_{f_1}, c_{f_2}, \dots, c_{f_p})$ тогаш

$$\vec{y} = c_{f_1} \vec{x}_1 + c_{f_2} \vec{x}_2 + \dots + c_{f_p} \vec{x}_p. \quad (\text{A.5'})$$

Потоа за секој кандидат за минимален пат вектор за ниво d се проверува дали сите линкови кои имаат ненегативен капацитет припаѓаат во $S_{\vec{x}}$. Ако припаѓаат во $S_{\vec{x}}$, тогаш кандидатот за минимален пат вектор е минимален пат вектор во спротивно не е

минимален пат вектор. Следниве две леми имаат значајна улога во определувањето дали некој кандидат за минимален пат вектор е минимален пат вектор.

Лема А.3 За секој кандидат за минимален пат вектор за ниво d постои најмалку еден минимален пат вектор за ниво d , \vec{y} така што $\vec{y} \leq \vec{x}$. Уште повеќе \vec{x} не е минимален пат вектор за ниво d ако постои вектор \vec{y} за кој е исполнето $\vec{y} < \vec{x}$.

Лема А.4 Ако транспортниот систем е ацикличен, тогаш секој кандидат за минимален пат вектор за ниво d е минимален пат вектор за ниво d .

Сега може да го дадеме алгоритмот предложен во [51].

Алгоритам А.1 - Алгоритам на Lin, Jane and Yuah (минимални патишта од s до $t : \mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_p$)

Чекор 1. За секој минимален пат \mathcal{P}_j ($j = 1, 2, \dots, p$) се наоѓа минималниот пат вектор за ниво 1, \vec{x}_j .

Чекор 2. Се наоѓаат индикаторите $c_{\vec{x}_j} = \min\{M_i \mid x_{j_i} = 1\}$.

Чекор 3. Се наоѓаат индикаторите $c_{\vec{f}} = (c_{f_1}, c_{f_2}, \dots, c_{f_p})$ согласно со равенствата (А.1) - (А.3) со методот на имплицитно најбројување:

$$\begin{aligned} \sum_{j=1}^p c_{f_j} &= d. \\ c_{f_j} &\geq c_{\vec{x}_j}, \quad j = 1, 2, \dots, p. \\ \sum_j \{c_{f_j} \mid x_{j_i} = 1\} &\leq M_i, \quad i = 1, 2, \dots, |E|. \end{aligned}$$

каде c_{f_j} е ненегативен цел број, за $j = 1, 2, \dots, |E|$.

Чекор 4. Трансформирај го секое решение во кандидат за минимален пат вектор за ниво d , $\vec{y} = (y_1, y_1, \dots, y_{|E|})$ со (А.5) или со (А.5')

$$y_i = \sum_j \{c_{f_j} \mid x_{f_j} = 1\} \leq M_i, \quad i = 1, 2, \dots, |E|$$

или

$$\vec{y} = c_{f_1} \vec{x}_1 + c_{f_2} \vec{x}_2 + \dots + c_{f_p} \vec{x}_p.$$

Чекор 5. За секој кандидат се проверува дали е минимален пат вектор за ниво d :

Чекор 5.1 Ако транспортниот систем е ацикличен, тогаш секој \vec{y} е минимален пат за ниво d .

Чекор 5.2 Ако транспортниот систем е цикличен, и $\vec{y}_1, \vec{y}_2, \dots, \vec{y}_\lambda$ се кандидати за минимални пат вектори, тогаш за секој кандидат се вршат долните чекори.

Чекор 5.2.1 $I = \emptyset$ (Во I се зачувуваат сите кандидати што не се минимални пат вектори).

Чекор 5.2.2 За секој $i = 1, 2, \dots, \lambda$

Чекор 5.2.3 За секој $j = 1, 2, \dots, \lambda$, $i \neq j, j \notin I$.

Чекор 5.2.4 Ако $\vec{y}_j < \vec{y}_i$ тогаш \vec{y}_i не е минимален пат вектор за ниво d , $I = I \cup \{i\}$ и оди на Чекор 5.2.7.

Чекор 5.2.5 Оди на следно j .

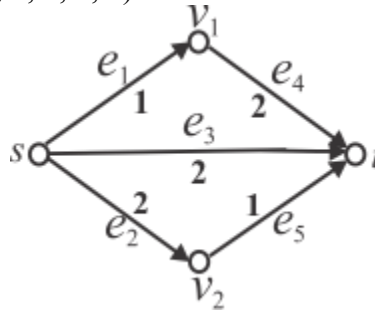
Чекор 5.2.6 \vec{y}_i е минимален пат вектор за ниво d .

Чекор 5.2.7 Оди на следно i .

Да ја најдеме сега временската сложеност на алгоритмот на Lin, Jane and Yuan.

Временската комплексност потребна за добивање на еден минимален пат вектор со (A.5) е $O(|E|^2)$. Поради тоа што претпоставивме дека λ е горната граница за бројот на добиени кандидати за минимални пат вектори временската сложеност на Чекор 4 е $O(\lambda|E|^2)$. Секој вектор \vec{y} има $|E|$ компоненти, па поради тоа временската комплексност за добивање на сите минимални пат вектори во Чекор 5 е $O(\lambda^2|E|)$. Поради тоа што $O(\lambda) \approx O(|E|^{|E|})$, за временската сложеност на алгоритмот се добива $O(\lambda|E|^2 + \lambda^2|E|) = O(\lambda^2|E|)$.

Пример А.1 Се разледува транспортниот систем даден на Фигура А.1. Максималните капацитети на линковите се $(1, 2, 2, 2, 1)$.



Фигура А.1 Насочен транспортен систем

Минималните патишта се $\mathcal{P}_1 = \{e_1, e_4\}$, $\mathcal{P}_2 = \{e_3\}$ и $\mathcal{P}_3 = \{e_2, e_5\}$.

Се бара ниво на работа на системот $d = 2$, фамилијата од минималните пат вектори за ниво 2 (2-MPs) е добиена на следниот начин:

Чекор 1. $1\text{-MP} = \{(1, 0, 0, 1, 0), (0, 0, 1, 0, 0), (0, 1, 0, 0, 1)\}$.

Чекор 2. $c_{\vec{x}_1} = \min\{1, 2\} = 1$, $c_{\vec{x}_2} = \min\{2\} = 2$, $c_{\vec{x}_3} = \min\{2, 1\} = 1$.

Чекор 3. Најди ги сите индикатори $c_{\vec{f}} = (c_{f_1}, c_{f_2}, c_{f_3})$ што ги задоволуваат ограничувањата (A.1) - (A.3) :

$$\begin{aligned}
 c_{f_1} + c_{f_2} + c_{f_3} &= 2, \\
 (c_{f_1}, c_{f_2}, c_{f_3}) &\leq (1, 2, 1), \\
 \sum_j \{c_{f_j} \mid x_{j_1} = 1\} &= c_{f_1} \leq 1,
 \end{aligned}$$

$$\begin{aligned}\sum_j \{c_{f_j} \mid x_{j_2} = 1\} &= c_{f_3} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_3} = 1\} &= c_{f_2} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_4} = 1\} &= c_{f_1} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_5} = 1\} &= c_{f_3} \leq 1.\end{aligned}$$

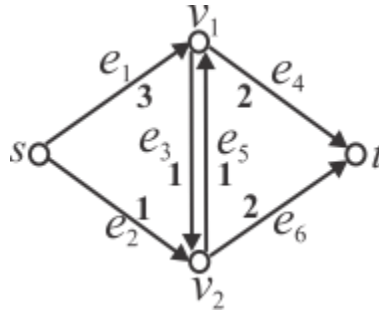
Ги добиваме следниве четири вектори $c_{\vec{f}_1} = (1, 1, 0)$, $c_{\vec{f}_2} = (1, 0, 1)$, $c_{\vec{f}_3} = (0, 1, 1)$ и $c_{\vec{f}_4} = (0, 2, 0)$.

Чекор 4. Со помош на $c_{\vec{f}_j}$ се добиваат 2-*MP* кандидатите $\vec{y} = (y_1, y_2, y_3, y_4, y_5)$:

$$\begin{aligned}\vec{y}_1 &= \vec{x}_1 + \vec{x}_2 = (1, 0, 0, 1, 0) + (0, 0, 1, 0, 0) = (1, 0, 1, 1, 0), \\ \vec{y}_2 &= \vec{x}_1 + \vec{x}_3 = (1, 0, 0, 1, 0) + (0, 1, 0, 0, 1) = (1, 1, 0, 1, 1), \\ \vec{y}_3 &= \vec{x}_2 + \vec{x}_3 = (0, 0, 1, 0, 0) + (0, 1, 0, 0, 1) = (0, 1, 1, 0, 1), \\ \vec{y}_4 &= 2\vec{x}_2 = 2(0, 0, 1, 0, 0) = (0, 0, 2, 0, 0).\end{aligned}$$

Чекор 5. Бидејќи транспортниот систем е ацикличен, секој кандидат е 2-*MP*. Значи $2\text{-}MP = \{(1, 0, 1, 1, 0), (1, 1, 0, 1, 1), (0, 1, 1, 0, 1), (0, 0, 2, 0, 0)\}$

Пример А.2 Се разгледува транспортниот систем од Фигура А.2. Максималните капацитети на линковите се $(3, 1, 1, 2, 1, 2)$.



Фигура А.2 Насочен транспортен систем

Минимални патишта се $\mathcal{P}_1 = \{e_1, e_4\}$, $\mathcal{P}_2 = \{e_1, e_3, e_6\}$, $\mathcal{P}_3 = \{e_2, e_5, e_4\}$ и $\mathcal{P}_4 = \{e_2, e_6\}$. Се бара ниво на работа на системот $d = 3$, фамилијата од минималните пат вектори за ниво 3 (3-*MPs*) е добиена на следниот начин:

Чекор 1. $1\text{-}MP = \{(1, 0, 0, 1, 0, 0), (1, 0, 1, 0, 0, 1), (0, 1, 0, 1, 1, 0), (0, 1, 0, 0, 0, 1)\}$

Чекор 2. $c_{\vec{x}_1} = \min\{3, 2\} = 2$, $c_{\vec{x}_2} = \min\{3, 1, 2\} = 1$, $c_{\vec{x}_3} = \min\{1, 1, 2\} = 1$ и $c_{\vec{x}_4} = \min\{1, 2\} = 1$

Чекор 3. Најди ги сите индикатори $c_{\vec{f}} = (c_{f_1}, c_{f_2}, c_{f_3}, c_{f_4})$ што ги задоволуваат ограничувањата (A.1) - (A.3) :

$$\begin{aligned} c_{f_1} + c_{f_2} + c_{f_3} + c_{f_4} &= 3, \\ (c_{f_1}, c_{f_2}, c_{f_3}, c_{f_4}) &\leq (2, 1, 1, 1), \\ \sum_j \{c_{f_j} \mid x_{j_1} = 1\} &= c_{f_1} + c_{f_2} \leq 3, \\ \sum_j \{c_{f_j} \mid x_{j_2} = 1\} &= c_{f_3} + c_{f_4} \leq 1, \\ \sum_j \{c_{f_j} \mid x_{j_3} = 1\} &= c_{f_2} \leq 1, \\ \sum_j \{c_{f_j} \mid x_{j_4} = 1\} &= c_{f_1} + c_{f_3} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_5} = 1\} &= c_{f_3} \leq 1, \\ \sum_j \{c_{f_j} \mid x_{j_6} = 1\} &= c_{f_2} + c_{f_4} \leq 2, \end{aligned}$$

Ги добиваме следниве четири вектори $c_{\vec{f}_1} = (2, 1, 0, 0)$, $c_{\vec{f}_2} = (1, 1, 1, 0)$, $c_{\vec{f}_3} = (2, 0, 0, 1)$ и $c_{\vec{f}_4} = (1, 1, 0, 1)$.

Чекор 4. Со помош на $c_{\vec{f}_j}$ се добиваат 3 – *MP* кандидатите $\vec{y} = (y_1, y_2, y_3, y_4, y_5, y_6)$:

$$\begin{aligned} \vec{y}_1 &= 2\vec{x}_1 + \vec{x}_2 = (2, 0, 0, 2, 0, 0) + (1, 0, 1, 0, 0, 1) = (3, 0, 1, 2, 0, 1), \\ \vec{y}_2 &= \vec{x}_1 + \vec{x}_2 + \vec{x}_3 = (1, 0, 0, 1, 0, 0) + (1, 0, 1, 0, 0, 1) + (0, 1, 0, 1, 1, 0) \\ &= (2, 1, 1, 2, 1, 1), \\ \vec{y}_3 &= 2\vec{x}_1 + \vec{x}_4 = (2, 0, 0, 2, 0, 0) + (0, 1, 0, 0, 0, 1) = (2, 1, 0, 2, 0, 1), \\ \vec{y}_4 &= \vec{x}_1 + \vec{x}_2 + \vec{x}_4 = (1, 0, 0, 1, 0, 0) + (1, 0, 1, 0, 0, 1) + (0, 1, 0, 0, 0, 1) \\ &= (2, 1, 1, 1, 0, 2) \end{aligned}$$

Чекор 5. Транспортниот систем е цикличен, и $\{\vec{y}_1, \vec{y}_2, \vec{y}_3, \vec{y}_4\}$ се 3-*MP* кандидати. Бидејќи $\vec{y}_3 = (2, 1, 0, 2, 0, 1) < \vec{y}_2 = (2, 1, 1, 2, 1, 1)$, \vec{y}_2 не е 3-*MP*.

Множеството од минимални пат вектори за ниво d , 3-*MP* = $\{(3, 0, 1, 2, 0, 1), (2, 1, 0, 2, 0, 1), (2, 1, 1, 1, 0, 2)\}$.

Во овој пример, 4 кандидати за минимални пат вектори се добиени, 3 се минимални вектори и 1 не е минимален пат вектор.

Пример А.3 Го разледуваме транспортниот систем од Фигура 5.1, која има 9 линкови и 6 јазли. Капацитетите на линковите се $(3, 2, 2, 1, 2, 1, 3, 1, 2)$ и се бара системот да работи со ниво $d = 3$.

Минималните патишта се $\mathcal{P}_1 = \{e_1, e_3, e_6, e_5, e_9\}$, $\mathcal{P}_2 = \{e_1, e_3, e_7\}$, $\mathcal{P}_3 = \{e_2, e_4, e_3, e_7\}$, $\mathcal{P}_4 = \{e_2, e_5, e_8, e_7\}$ и $\mathcal{P}_5 = \{e_2, e_5, e_9\}$. Се бара ниво на работа на системот $d = 3$, фамилијата од минималните пат вектори за ниво 3, $3\text{-}MPs$ е добиена на следниот начин:

Чекор 1. $1\text{-}MP = \{(1, 0, 1, 0, 1, 1, 0, 0, 1), (1, 0, 1, 0, 0, 0, 1, 0, 0), (0, 1, 1, 1, 0, 0, 1, 0, 0), (0, 1, 0, 0, 1, 0, 1, 1, 0), (0, 1, 0, 0, 1, 0, 0, 0, 1)\}$

Чекор 2. $c_{\bar{x}_1} = \min\{3, 2, 1, 2, 2\} = 1$, $c_{\bar{x}_2} = \min\{3, 2, 3\} = 2$, $c_{\bar{x}_3} = \min\{2, 1, 2, 3\} = 1$,
 $c_{\bar{x}_4} = \min\{2, 2, 1, 3\} = 1$ и $c_{\bar{x}_5} = \min\{2, 2, 2\} = 2$.

Чекор 3. Најди ги сите вектори $c_{\bar{f}} = (c_{f_1}, c_{f_2}, c_{f_3}, c_{f_4}, c_{f_5})$ кои ги задоволуваат ограничувањата (A.1) - (A.3) :

$$\begin{aligned} c_{f_1} + c_{f_2} + c_{f_3} + c_{f_4} + c_{f_5} &= 3, \\ (c_{f_1}, c_{f_2}, c_{f_3}, c_{f_4}, c_{f_5}) &\leq (1, 2, 1, 1, 2), \\ \sum_j \{c_{f_j} \mid x_{j_1} = 1\} &= c_{f_1} + c_{f_2} \leq 3, \\ \sum_j \{c_{f_j} \mid x_{j_2} = 1\} &= c_{f_3} + c_{f_4} + c_{f_5} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_3} = 1\} &= c_{f_1} + c_{f_2} + c_{f_3} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_4} = 1\} &= c_{f_3} \leq 1, \\ \sum_j \{c_{f_j} \mid x_{j_5} = 1\} &= c_{f_1} + c_{f_4} + c_{f_5} \leq 2, \\ \sum_j \{c_{f_j} \mid x_{j_6} = 1\} &= c_{f_1} \leq 1, \\ \sum_j \{c_{f_j} \mid x_{j_7} = 1\} &= c_{f_2} + c_{f_3} + c_{f_4} \leq 3, \\ \sum_j \{c_{f_j} \mid x_{j_8} = 1\} &= c_{f_4} \leq 1, \\ \sum_j \{c_{f_j} \mid x_{j_9} = 1\} &= c_{f_1} + c_{f_5} \leq 2, \end{aligned}$$

Добиени се следниве седум вектори $c_{\bar{f}_1} = (0, 1, 0, 1, 1)$, $c_{\bar{f}_2} = (0, 1, 1, 0, 1)$, $c_{\bar{f}_3} = (0, 1, 1, 1, 0)$, $c_{\bar{f}_4} = (1, 0, 1, 1, 0)$, $c_{\bar{f}_5} = (1, 1, 0, 1, 0)$, $c_{\bar{f}_6} = (0, 2, 0, 0, 1)$, $c_{\bar{f}_7} = (0, 2, 0, 1, 0)$,
 $c_{\bar{f}_8} = (0, 1, 0, 0, 2)$, $c_{\bar{f}_9} = (1, 1, 0, 0, 1)$ и $c_{\bar{f}_{10}} = (1, 0, 1, 0, 1)$.

Чекор 4. Со помош на $c_{\bar{f}_j}$ се добиваат кандидатите $\bar{y} = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9)$:

$$\begin{aligned} \bar{y}_1 &= \bar{x}_1 + \bar{x}_4 + \bar{x}_5 = (1, 2, 1, 0, 2, 0, 2, 1, 1), \\ \bar{y}_2 &= \bar{x}_2 + \bar{x}_3 + \bar{x}_5 = (1, 2, 2, 1, 1, 0, 2, 0, 1), \end{aligned}$$

$$\begin{aligned}
\bar{y}_3 &= \bar{x}_2 + \bar{x}_3 + \bar{x}_4 = (1, 2, 2, 1, 1, 0, 3, 1, 0), \\
\bar{y}_4 &= \bar{x}_1 + \bar{x}_3 + \bar{x}_4 = (1, 2, 2, 1, 2, 1, 2, 1, 1), \\
\bar{y}_5 &= \bar{x}_1 + \bar{x}_2 + \bar{x}_4 = (2, 1, 2, 0, 2, 1, 2, 1, 1), \\
\bar{y}_6 &= 2\bar{x}_2 + \bar{x}_5 = (2, 1, 2, 0, 1, 0, 2, 0, 1), \\
\bar{y}_7 &= 2\bar{x}_2 + \bar{x}_4 = (2, 1, 2, 0, 1, 0, 3, 1, 0), \\
\bar{y}_8 &= \bar{x}_2 + 2\bar{x}_5 = (1, 2, 1, 0, 2, 0, 1, 0, 2), \\
\bar{y}_9 &= \bar{x}_1 + \bar{x}_2 + \bar{x}_5 = (2, 1, 2, 0, 2, 1, 1, 0, 2), \\
\bar{y}_{10} &= \bar{x}_1 + \bar{x}_3 + \bar{x}_5 = (1, 2, 2, 1, 2, 1, 1, 0, 2).
\end{aligned}$$

Чекор 5. Транспортниот систем е цикличен, и $\{\bar{y}_1, \bar{y}_2, \bar{y}_3, \bar{y}_4, \bar{y}_5, \bar{y}_6, \bar{y}_7, \bar{y}_8, \bar{y}_9, \bar{y}_{10}\}$ се 3-*MP* кандидати. Бидејќи $\bar{y}_2 = (1, 2, 2, 1, 1, 0, 2, 0, 1) < \bar{y}_4 = (1, 2, 2, 1, 2, 1, 2, 1, 1)$, $\bar{y}_6 = (2, 1, 2, 0, 1, 0, 2, 0, 1) < \bar{y}_5 = (2, 1, 2, 0, 2, 1, 2, 1, 1)$ и $\bar{y}_8 = (1, 2, 1, 0, 2, 0, 1, 0, 2) < \bar{y}_{10} = (1, 2, 2, 1, 2, 1, 1, 0, 2)$, \bar{y}_4 , \bar{y}_5 и \bar{y}_{10} не се 3-*MP*.

Множеството од минимални пат вектори за ниво 3-*MP* = $\{(1, 2, 1, 0, 2, 0, 2, 1, 1), (1, 2, 2, 1, 1, 0, 2, 0, 1), (1, 2, 2, 1, 1, 0, 3, 1, 0), (2, 1, 2, 0, 1, 0, 2, 0, 1), (2, 1, 2, 0, 1, 0, 3, 1, 0), (1, 2, 1, 0, 2, 0, 1, 0, 2), (2, 1, 2, 0, 2, 1, 1, 0, 2)\}$.

Во овој пример, 8 кандидати за минимални пат вектори се, 6 се минимални пат вектори и 2 не се минимални пат вектори.

Уште да напоменеме дека во трудот на Y. K. Lin [60] е покажано дека ако важи равенството (A.3) тогаш важи и неравенството (A.2.). За да ги најде сите минимални пат вектори, прво ги наоѓаа решенијата на (A.1) и (A.3) и потоа го трансформираат секој вектор до кандидат за минимал пат вектор за ниво d , со користење на (A.5) или (A.5').

Наредно ќе го разгледаме алгоритмот за добивање на минимални пат вектори предложен од M. Forghani-elahabad и N. Mahdavi-Amiri. Овој алгоритам всушност е подобрување на предходниот алгоритам предложен од Lin, Jane и Yuan. Овие алгоритми се разликуваат само во последниот чекор, Чекор 5. Кај алгоритмот на Lin кандидатите за минимални пат вектори што не се минимални се елиминираат со взаемна споредба на сите кандидати, додека во алгоритмот на M. Forghani-elahabad и N. Mahdavi-Amiri е предложен поефикасен начин за нивно елиминирање со користење на алгоритам за максимален проток [47].

Всушност, M. Forghani-elahabad и N. Mahdavi-Amiri проверуваат за секој кандидат за минимален пат вектор \bar{x} , дали постои линк $e_i \in N_{\bar{x}}$ за кој важи $\varphi(\bar{x} - \bar{a}_i) = d$, каде $\bar{a}_i = (\delta_{i1}, \delta_{i1}, \dots, \delta_{i|E|})$, со $\delta_{ij} = 1$ ако $i = j$ и $\delta_{ij} = 0$ ако $i \neq j$. Ако постои таков линк, тогаш кандидатот за минимален пат вектор за ниво d , \bar{x} не е минимален пат вектор за ниво d . Ова е јасно, бидејќи доколку постои вектор $\bar{x} - \bar{a}_i < \bar{x}$, кој е пат вектор за ниво d , тогаш тој е пат вектор што е помал од \bar{x} и јасно \bar{x} не е минимален пат вектор.

Ќе го дадеме само чекорот во кој се разликуваат овие алгоритми, како што е предходно кажавмо разликата се состои само во последниот чекор

Чекор 5. $Q = \{\vec{y}_1, \vec{y}_2, \dots, \vec{y}_\lambda\}$ е множество од кандидати за минимални пат вектори за ниво d . За секој кандидат се проверува дали е минимален пат вектор на следноиот начин:

За секој $j = 1, 2, \dots, \lambda$

Ако постои $e_i \in N_{\vec{y}_j}$ за кој $\varphi(\vec{y}_j - \vec{a}_i) = d$

тогаш \vec{y}_j не е $d - MP$ и се брише од Q .

Да ја разгледаме временската сложеност на алгоритмот на М. Forghani-elahabad и N. Mahdavi-Amiri. Во Чекорот 5, алгоритмот за максимален проток се користи најмногу $|E|$ пати (имаме $|E|$ - торки) за секој кандидат за минимален пат вектор \vec{y}_j . Комплексноста на алгоритмот за максимален пресек е $O(|V|^2|E|^{1/2})$ [47]. Бидејќи имаме λ кандидати за минимални пат вектори, (секој од нив е $|E|$ - торка) комплексноста во Чекор 5 ќе биде $O(|V|^2|E|^{3/2}\lambda)$. За предходниот алгоритам разледаваме дека временската сложеност за добивање на сите кандидати за минимални пат вектори е $O(\lambda|E|^2)$. Па според ова временската сложеност на алгоритмот предложен од М. Forghani-elahabad и N. Mahdavi-Amiri е $O(|E|^2\lambda) + |V|^2|E|^{3/2}\lambda = O(|V|^2|E|^{3/2}\lambda)$. За сложеност на алгоритмот предложен од Lin, Jane и Yuan е добиено $O(|E|\lambda^2)$. Бидејќи $O(\lambda) \approx O(|E|^{E/})$ и $O(|V|) < O(|E|) < O(|V|^2)$, следува дека $|V|^2|E|^{3/2} < \lambda$ и $O(|V|^2|E|^{3/2}\lambda) < O(|E|\lambda^2)$. Од каде имаме дека алгоритмот на М. Forghani-elahabad и N. Mahdavi-Amiri е поефикасен од алгоритмот на Lin, Jane и Yuan.

Пример А.4 Да го примениме алгоритмот на М. Forghani-elahabad и N. Mahdavi-Amiri на Пример А.2. Со примена на алгоритмот на Lin, го добивме множеството од кандидати за минимални пат вектори за ниво 3. $Q = \{(3, 0, 1, 2, 0, 1), (2, 1, 1, 2, 1, 1), (2, 1, 0, 2, 0, 1), (2, 1, 1, 1, 0, 2)\}$.

Чекор 5. За секој елемент од Q се проверува дали постои $e_i \in N_{\vec{y}_j}$ така што $\varphi(\vec{y}_j - \vec{a}_i) = 3$

$$\vec{y}_1 = (3, 0, 1, 2, 0, 1),$$

$$\varphi(\vec{y}_1 - \vec{a}_1) = \varphi((2, 0, 1, 2, 0, 1)) = 2 < 3,$$

$$\varphi(\vec{y}_1 - \vec{a}_3) = \varphi((3, 0, 0, 2, 0, 1)) = 2 < 3,$$

$$\varphi(\vec{y}_1 - \vec{a}_4) = \varphi((3, 0, 1, 1, 0, 1)) = 2 < 3,$$

$$\varphi(\vec{y}_1 - \vec{a}_6) = \varphi((3, 0, 1, 2, 0, 0)) = 2 < 3,$$

Значи \vec{y}_1 е минимален пат вектор за ниво 3.

$$\vec{y}_2 = (2, 1, 1, 2, 1, 1),$$

$$\varphi(\vec{y}_2 - \vec{a}_1) = \varphi((1, 1, 1, 2, 1, 1)) = 2 < 3,$$

$$\varphi(\vec{y}_2 - \vec{a}_2) = \varphi((2, 0, 1, 2, 1, 1)) = 2 < 3,$$

$$\varphi(\vec{y}_2 - \vec{a}_3) = \varphi((2, 1, 0, 2, 1, 1)) = 3,$$

Значи \vec{y}_2 не е минимален пат вектор за ниво 3.

$$\vec{y}_3 = (2, 1, 0, 2, 0, 1),$$

$$\varphi(\vec{y}_3 - \vec{a}_1) = \varphi((1, 1, 0, 2, 0, 1)) = 2 < 3$$

$$\varphi(\vec{y}_3 - \vec{a}_2) = \varphi((2, 0, 0, 2, 0, 1)) = 2 < 3$$

$$\varphi(\vec{y}_3 - \vec{a}_4) = \varphi((2, 1, 0, 1, 0, 1)) = 2 < 3$$

$$\varphi(\vec{y}_3 - \vec{a}_6) = \varphi((2, 1, 0, 2, 0, 0)) = 2 < 3$$

Значи \vec{y}_3 е минимален пат вектор за ниво 3.

$$\vec{y}_4 = (2, 1, 1, 1, 0, 2)$$

$$\varphi(\vec{y}_4 - \vec{a}_1) = \varphi((1, 1, 1, 1, 0, 2)) = 2 < 3$$

$$\varphi(\vec{y}_4 - \vec{a}_2) = \varphi((2, 0, 1, 1, 0, 2)) = 2 < 3$$

$$\varphi(\vec{y}_4 - \vec{a}_3) = \varphi((2, 1, 0, 1, 0, 2)) = 2 < 3$$

$$\varphi(\vec{y}_4 - \vec{a}_4) = \varphi((2, 1, 1, 0, 0, 2)) = 2 < 3$$

$$\varphi(\vec{y}_4 - \vec{a}_6) = \varphi((2, 1, 1, 1, 0, 1)) = 2 < 3$$

Значи \vec{y}_4 е минимален пат вектор за ниво 3.

Множеството од минимални пат вектори за ниво d , $3\text{-}MP = \{(3, 0, 1, 2, 0, 1), (2, 1, 0, 2, 0, 1), (2, 1, 1, 1, 0, 2)\}$.

Пример А.5 Да го примениме алгоритмот на М. Forghani-elahabad и N. Mahdavi-Amiri на Пример А.3. Со примена на алгоритмот на Lin, го добивме множеството од кандидати за минимални пат вектори за ниво 3. $Q = \{(1, 2, 1, 0, 2, 0, 2, 1, 1), (1, 2, 2, 1, 1, 0, 2, 0, 1), (1, 2, 2, 1, 1, 0, 3, 1, 0), (1, 2, 2, 1, 2, 1, 2, 1, 1), (2, 1, 2, 0, 2, 1, 2, 1, 1), (2, 1, 2, 0, 1, 0, 2, 0, 1), (2, 1, 2, 0, 1, 0, 3, 1, 0), (1, 2, 1, 0, 2, 0, 1, 0, 2), (2, 1, 2, 0, 2, 1, 1, 0, 2), (1, 2, 2, 1, 2, 1, 1, 0, 2)\}$ се $3\text{-}MP$ кандидати.

Чекор 5. За секој елемент од Q се проверува дали постои $e_i \in N_{\vec{y}_j}$ така што $\varphi(\vec{y}_j - \vec{a}_i) = 3$

$$\vec{y}_1 = (1, 2, 1, 0, 2, 0, 2, 1, 1),$$

$$\varphi(\vec{y}_1 - \vec{a}_i) = 2 < 3 \text{ за } i = 1, 2, 3, 5, 7, 8 \text{ и } 9,$$

$\Rightarrow \vec{y}_1$ е минимален пат вектор за ниво 3.

$$\vec{y}_2 = (1, 2, 2, 1, 1, 0, 2, 0, 1),$$

$$\varphi(\vec{y}_2 - \vec{a}_i) = 2 < 3 \text{ за } i = 1, 2, 3, 4, 5, 7 \text{ и } 9,$$

$\Rightarrow \vec{y}_2$ е минимален пат вектор за ниво 3.

$$\vec{y}_3 = (1, 2, 2, 1, 1, 0, 3, 1, 0),$$

$$\varphi(\vec{y}_3 - \vec{a}_i) = 2 < 3 \text{ за } i = 1, 2, 3, 4, 5, 7 \text{ и } 8,$$

$\Rightarrow \vec{y}_3$ е минимален пат вектор за ниво 3.

$$\vec{y}_4 = (1, 2, 2, 1, 2, 1, 2, 1, 1),$$

$$\varphi(\vec{y}_4 - \vec{a}_4) = 3,$$

$\Rightarrow \vec{y}_4$ не е минимален пат вектор за ниво 3.

$$\vec{y}_5 = (2, 1, 2, 0, 2, 1, 2, 1, 1),$$

$$\varphi(\vec{y}_5 - \vec{a}_6) = 3,$$

$\Rightarrow \vec{y}_5$ не е минимален пат вектор за ниво 3.

$$\vec{y}_6 = (2, 1, 2, 0, 1, 0, 2, 0, 1),$$

$$\varphi(\vec{y}_6 - \vec{a}_i) = 2 < 3 \text{ за } i = 1, 2, 3, 5, 7 \text{ и } 9,$$

$\Rightarrow \vec{y}_6$ е минимален пат вектор за ниво 3.

$$\vec{y}_7 = (2, 1, 2, 0, 1, 0, 3, 1, 0),$$

$$\varphi(\vec{y}_7 - \vec{a}_i) = 2 < 3 \text{ за } i = 1, 2, 3, 5, 7 \text{ и } 8,$$

$\Rightarrow \vec{y}_7$ е минимален пат вектор за ниво 3.

$$\vec{y}_8 = (1, 2, 1, 0, 2, 0, 1, 0, 2),$$

$$\varphi(\vec{y}_8 - \vec{a}_i) = 2 < 3 \text{ за } i = 1, 2, 3, 5, 7 \text{ и } 9,$$

$\Rightarrow \vec{y}_8$ е минимален пат вектор за ниво 3.

$$\vec{y}_9 = (2, 1, 2, 0, 2, 1, 1, 0, 2),$$

$$\varphi(\vec{y}_9 - \vec{a}_i) = 2 < 3 \text{ за } i = 1, 2, 3, 5, 6, 7 \text{ и } 9,$$

$\Rightarrow \vec{y}_9$ е минимален пат вектор за ниво.

$$\vec{y}_{10} = (1, 2, 2, 1, 2, 1, 1, 0, 2),$$

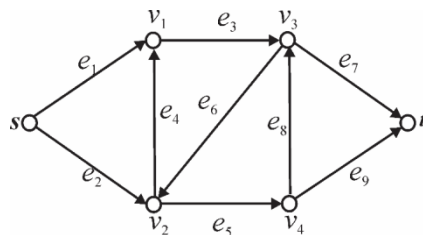
$$\varphi(\vec{y}_{10} - \vec{a}_3) = 3,$$

$\Rightarrow \vec{y}_{10}$ не е минимален пат вектор за ниво 3.

Множеството од минимални пат вектори за ниво 3- $MP = \{(1, 2, 1, 0, 2, 0, 2, 1, 1), (1, 2, 2, 1, 1, 0, 2, 0, 1), (1, 2, 2, 1, 1, 0, 3, 1, 0), (2, 1, 2, 0, 1, 0, 2, 0, 1), (2, 1, 2, 0, 1, 0, 3, 1, 0), (1, 2, 1, 0, 2, 0, 1, 0, 2), (2, 1, 2, 0, 2, 1, 1, 0, 2), \}$.

Да ги разгледаме сега кандидатите за минимални пат вектори за ниво 3 кои не се минимални. Тоа се векторите $(1, 2, 2, 1, 2, 1, 2, 1, 1)$, $(2, 1, 2, 0, 2, 1, 2, 1, 1)$ и $(1, 2, 2, 1, 2, 1, 1, 0, 2)$. Да видиме што може да забележиме ако ги најцртаме графовите за овие кандидати за минимални пат вектори.

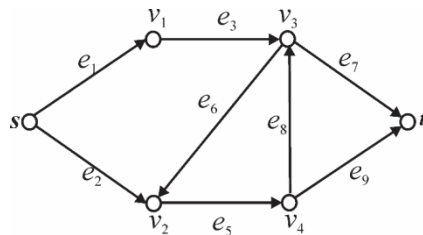
1. $(1, 2, 2, 1, 2, 1, 1, 0, 2)$



Фигура A.4 Пат вектор за ниво 3, (1, 2, 2, 1, 2, 1, 1, 0, 2)

Да забележеме дека овој пат вектор ги содржи циклусите $\langle v_2, v_1, v_3, v_2 \rangle$ и $\langle v_2, v_4, v_3, v_2 \rangle$.

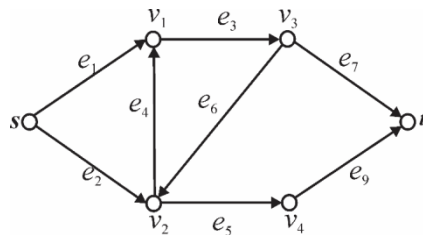
2. (2, 1, 2, 0, 2, 1, 2, 1, 1)



Фигура A.5 Пат вектор за ниво 3, (2, 1, 2, 0, 2, 1, 2, 1, 1)

Овој пат вектор го содржи циклусот $\langle v_2, v_4, v_3, v_2 \rangle$.

3. (1, 2, 2, 1, 2, 1, 1, 0, 2)



Фигура A.6 Пат вектор за ниво 3, (1, 2, 2, 1, 2, 1, 1, 0, 2)

Овој пат вектор го содржи циклусот $\langle v_2, v_4, v_3, v_2 \rangle$.

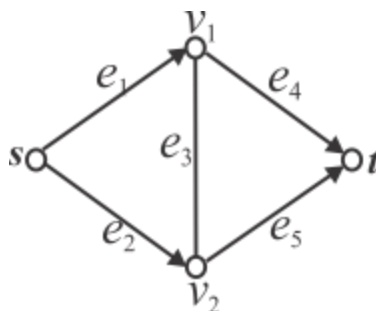
Можеме да заклучиме дека оние кандидати за минимални пат вектори, кои содржат циклуси не се минимални пат вектори. Заради ова согледуваме во Глава 5 се предложени алгоритми за наоѓање на минимални пат вектори каде што за секој кандидат за минимален вектор ќе вршиме проверка дали векторот содржи циклуси. Доколку содржи, во тој случај веднаш заклучуваме дека тој кандидат не е минимален пат вектор.

Прилог Б

Алгоритам за наоѓање на минимални пат вектори предложен од М. Mihova и N. Synagina

Се разгледува ненасочен транспортен систем $G(V, E, c)$ со множество од јазли $V = \{v_1, v_2, \dots, v_{|V|}\}$, множество од линкови $E = \{e_1, e_2, \dots, e_{|E|}\}$, c е функција од капацитет. Да претпоставиме дека i - от линк може да работи во едно од нивоата од множеството $\{0, 1, \dots, M_i\}$, додека максималниот проток што може да се пренесе низ транспортниот систем да го означеме со M . ($\{0, 1, \dots, M\}$ е множество од сите можно протоци)

Веднаш да кажеме дека овој алгоритам работи и за нанасочени и за ненасочени транспортни системи, за разлика од претходните два алгоритми кои работат само за насочени транспортни системи. Исто така со овој алгоритам се добиваат сите минималните пат вектори тргнувајќи од минималните пат вектори за ниво 1 до минималните пат вектори за максималното ниво M . И овде претпоставуваме дека минимални патишта на транспортниот систем, $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_p$, се познати однапред. Од минималните патишта може да се најдат минимални пат вектори за ниво 1. Тоа се всушност минималните пат вектори за бинарниот транспортен систем кој ја има истата структура како и повеќе-состојбниот транспортен систем. Да го разгледаме транспортниот систем прикажан на Фигура Б.1.



Фигура Б.1 Насочен транспортен систем

Минимални патишта за бинарниот транспортниот систем претставен на Фигура Б.1 се $\mathcal{P}_1 = \{e_1, e_4\}$, $\mathcal{P}_2 = \{e_1, e_3, e_5\}$, $\mathcal{P}_3 = \{e_2, e_3, e_4\}$ и $\mathcal{P}_4 = \{e_2, e_5\}$. Ако сакаме да го разледуваме системот во повеќе-состојбен контекст, минималните пат вектори се: $(1, 0, 0, 1, 0)$, $(1, 0, 1, 0, 1)$, $(0, 1, 1, 1, 0)$ и $(0, 1, 0, 0, 1)$ и ова се минимални пат вектори за ниво 1.

Ако еден линк е двонасочен, тогаш ќе избереме едната насока да биде позитивна и земаме 1 за оваа насока и -1 за другата насока. На овој начин за секој минимален пат, земаме нов вектор **минимален насочен пат вектор (MDP)**. Во транспортниот систем прикажан на Фигура Б.1 линкот e_3 е двонасочен, и во овој случај ќе ја избереме насоката од v_1 до v_2 за позитивна насока и спротивната насока за негативна насока. Според ова минималните насочен пат вектори се : $(1, 0, 0, 1, 0)$, $(1, 0, 1, 0, 1)$, $(0, 1, -1, 1, 0)$ и $(0, 1, 0, 0, 1)$.

Дефиниција Б.1 Нека \vec{x} е минимален пат вектор за ниво d ($d - MP$), тогаш минималниот насочен пат вектор за ниво d ($d - MDP$), се добива кога во \vec{x} , x_i се заменува со $-x_i$, секогаш кога i - от линк е двонасочен и се користи негативната насока.

Сега можеме да го дадеме псевдокодот на алгоритмот предложен од М. Mihova и N.Synagina.

Алгоритам Б.2 - Алгоритам на Mihova and Synagina (минимални патишта од s до t $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_p$)

Чекор 1. За секој минимален пат \mathcal{P}_j ($j = 1, 2, \dots, p$) се наоѓа минималниот пат вектор за ниво 1.

Чекор 2. За секоја минимален пат вектор за ниво 1 се наоѓа минималниот насочен пат вектор за ниво 1.

Чекор 3. Се конструира множеството $(d+1) - MDP' = \{\vec{x} + \vec{y} \mid \vec{x} \in d - MDP \text{ и } \vec{y} \in 1 - MDP\}$.

Чекор 4. Се наоѓа $Abs(\vec{x}) = (|x_1|, |x_2|, \dots, |x_E|)$.

Чекор 5. Множеството $d - MDP$ е добиено од множеството $d - MDP'$ со елиминирање на елементи што се појавуваат повеќе од еднаш (дупликат елементи) и елиминирање на сите кандидати во кои постојат линкови низ кои поминува проток што е поголем од капацитетот на соодветниот линк ($> M_i$).

Чекор 6. Множеството $d - MP$ е добиено од множеството $\{Abs(\vec{x}) \mid \vec{x} \in d - MDP\}$ со елиминирање на сите елементи што се појавуваат повеќе од еднаш и елиминирање на сите елементи \vec{x} за кои постои елемент \vec{y} , т.ш. $\vec{x} > \vec{y}$.

Чекор 7. Повтори ги чекорите 3, 4, 5 и 6 за сите $d = 1, 2, \dots, M$.

Лема Б.3 Нека двотерминалниот транспортен систем има μ минимални пат вектори за ниво 1. Тогаш бројот од множества што содржат d минимални пат вектори за ниво 1 е еднаков на:

$$\binom{\mu + d - 1}{d}. \quad (\text{Б.1})$$

Доказот на лемата следува директно од фактот дека постојат точно $\binom{\mu + d - 1}{d}$ начини да

се распределат d елементи во μ класи (комбинации со повторување).

За да се добијат сите минимални пат вектори за ниво d , треба да ги пресметаме сите суми

$\vec{x} = \sum_{k=1}^d \vec{x}_k, \vec{x}_k \in 1-MDP$. Бидејќи бројот на вакви суми е $\binom{\mu+d-1}{d}$ и $O\left(\binom{\mu+d-1}{d}\right) = O(\mu^d)$. Според тоа ја имаме следнава теорема.

Теорема Б.1 Нека двотерминалниот транспортен систем има μ минимални пат вектори за ниво 1. Тогаш временската сложеност на алгоритмот за директно определување на минималното пат множество за ниво d е помала од $O(\mu^d)$.

Пример Б.1 Повторно ја разледуваме мрежата прикажана на Фигура А.3. Во примерот сите линкови ни се насочени па затоа множествата од сите минимални пат вектори и сите минимални насочени пат вектори се еднакви. Капацитетите на линковите се (3, 2, 2, 1, 2, 1, 3, 1, 2) и максималното ниво на работа на транспортниот систем е 4.

Минималните патишта се $\mathcal{P}_1 = \{e_1, e_3, e_6, e_5, e_9\}$, $\mathcal{P}_2 = \{e_1, e_3, e_7\}$, $\mathcal{P}_3 = \{e_2, e_4, e_3, e_7\}$, $\mathcal{P}_4 = \{e_2, e_5, e_8, e_7\}$ и $\mathcal{P}_5 = \{e_2, e_5, e_9\}$.

1-MP = $\{\vec{x}_1 = (1, 0, 1, 0, 1, 1, 0, 0, 1), \vec{x}_2 = (1, 0, 1, 0, 0, 0, 1, 0, 0), \vec{x}_3 = (0, 1, 1, 1, 0, 0, 1, 0, 0), \vec{x}_4 = (0, 1, 0, 0, 1, 0, 1, 1, 0), \vec{x}_5 = (0, 1, 0, 0, 1, 0, 0, 0, 1)\}$.

Во Табела Б.1, Табела Б.2 и Табела Б.3 нема да ги прикажеме оние кандидати за минимални пат вектори што го надминуваат максималниот капацитет, како и оние што се појавуваат повеќе од еднаш. За останатите кандидати е дадено како се добиени.

Табела Б.1 Минимални пат вектори за ниво 2.

\vec{y}_i	2-MP кандидати	2-MP
\vec{y}_1	$\vec{x}_1 + \vec{x}_2 = (2, 0, 2, 0, 1, 1, 1, 0, 1)$	$(2, 0, 2, 0, 1, 1, 1, 0, 1)$
\vec{y}_2	$\vec{x}_1 + \vec{x}_3 = (1, 1, 2, 1, 1, 1, 1, 0, 1)$	$(1, 1, 2, 1, 1, 1, 1, 0, 1)$
\vec{y}_3	$\vec{x}_1 + \vec{x}_4 = (1, 1, 1, 0, 2, 1, 1, 1, 1)$	$(1, 1, 1, 0, 2, 1, 1, 1, 1)$
\vec{y}_4	$\vec{x}_1 + \vec{x}_5 = (1, 1, 1, 0, 2, 1, 0, 0, 2)$	$(1, 1, 1, 0, 2, 1, 0, 0, 2)$
\vec{y}_5	$\vec{x}_2 + \vec{x}_2 = (2, 0, 2, 0, 0, 0, 2, 0, 0)$	$(2, 0, 2, 0, 0, 0, 2, 0, 0)$
\vec{y}_6	$\vec{x}_2 + \vec{x}_3 = (1, 1, 2, 1, 0, 0, 2, 0, 0)$	$(1, 1, 2, 1, 0, 0, 2, 0, 0)$
\vec{y}_7	$\vec{x}_2 + \vec{x}_4 = (1, 1, 1, 0, 1, 0, 2, 1, 0)$	$(1, 1, 1, 0, 1, 0, 2, 1, 0)$
\vec{y}_8	$\vec{x}_2 + \vec{x}_5 = (1, 1, 1, 0, 1, 0, 1, 0, 1)$	$(1, 1, 1, 0, 1, 0, 1, 0, 1)$
\vec{y}_9	$\vec{x}_3 + \vec{x}_4 = (0, 2, 1, 1, 1, 0, 2, 1, 0)$	$(0, 2, 1, 1, 1, 0, 2, 1, 0)$
\vec{y}_{10}	$\vec{x}_3 + \vec{x}_5 = (0, 2, 1, 1, 1, 0, 1, 0, 1)$	$(0, 2, 1, 1, 1, 0, 1, 0, 1)$
\vec{y}_{11}	$\vec{x}_4 + \vec{x}_5 = (0, 2, 0, 0, 2, 0, 1, 1, 1)$	$(0, 2, 0, 0, 2, 0, 1, 1, 1)$
\vec{y}_{11}	$\vec{x}_5 + \vec{x}_5 = (0, 2, 0, 0, 2, 0, 0, 0, 2)$	$(0, 2, 0, 0, 2, 0, 0, 0, 2)$

Можеме да забележиме дека сите кандидати за минимални пат вектори за ниво 2 се минимални пат вектори за ниво 2.

Табела Б.2 Минимални пат вектори за ниво 3.

\vec{z}_i	3- <i>MP</i> кандидати	3- <i>MP</i>
\vec{z}_1	$\vec{y}_1 + \vec{x}_4 = (2, 1, 2, 0, 2, 1, 2, 1, 1)$	не е 3- <i>MP</i> , ($\vec{z}_1 > \vec{z}_6$)
\vec{z}_2	$\vec{y}_1 + \vec{x}_5 = (2, 1, 2, 0, 2, 1, 1, 0, 2)$	(2, 1, 2, 0, 2, 1, 1, 0, 2)
\vec{z}_3	$\vec{y}_2 + \vec{x}_4 = (1, 2, 2, 1, 2, 1, 2, 1, 1)$	не е 3- <i>MP</i> ($\vec{z}_3 > \vec{z}_9$)
\vec{z}_4	$\vec{y}_2 + \vec{x}_5 = (1, 2, 2, 1, 2, 1, 1, 0, 2)$	не е 3- <i>MP</i> ($\vec{z}_4 > \vec{z}_{10}$)
\vec{z}_5	$\vec{y}_5 + \vec{x}_4 = (2, 1, 2, 0, 1, 0, 3, 1, 0)$	(2, 1, 2, 0, 1, 0, 3, 1, 0)
\vec{z}_6	$\vec{y}_5 + \vec{x}_5 = (2, 1, 2, 0, 1, 0, 2, 0, 1)$	(2, 1, 2, 0, 1, 0, 2, 0, 1)
\vec{z}_7	$\vec{y}_6 + \vec{x}_4 = (1, 2, 2, 1, 1, 0, 3, 1, 0)$	(1, 2, 2, 1, 1, 0, 3, 1, 0)
\vec{z}_8	$\vec{y}_6 + \vec{x}_5 = (1, 2, 2, 1, 1, 0, 2, 0, 1)$	(1, 2, 2, 1, 1, 0, 2, 0, 1)
\vec{z}_9	$\vec{y}_{11} + \vec{x}_5 = (1, 2, 1, 0, 2, 0, 2, 1, 1)$	(1, 2, 1, 0, 2, 0, 2, 1, 1)
\vec{z}_{10}	$\vec{y}_{12} + \vec{x}_5 = (1, 2, 1, 0, 2, 0, 1, 0, 2)$	(1, 2, 1, 0, 2, 0, 1, 0, 2)

Табела Б.3 Минимални пат вектори за ниво 4.

\vec{w}_i	4- <i>MP</i> кандидати	4- <i>MP</i>
\vec{w}_1	$\vec{z}_5 + \vec{x}_5 = (2, 2, 2, 0, 2, 0, 3, 1, 1)$	(2, 2, 2, 0, 2, 0, 3, 1, 1)
\vec{w}_2	$\vec{z}_6 + \vec{x}_5 = (2, 2, 2, 0, 2, 0, 2, 0, 2)$	(2, 2, 2, 0, 2, 0, 2, 0, 2)

Можеме да забележиме дека ги добиваме истите кандидати за минимални пат вектори за ниво 3 како и во предходните Пример А.4 и Пример А.5. Повторно можеме да забележиме дека кандидатите за минимални пат вектори, кои содржат циклуси не се минимални пат вектори.